THERMODYNAMIC ANALYSIS AND OPTIMIZATION OF A NEW AMMONIA BASED COMBINED POWER/COOLING CYCLE

By

SHAOGUANG LU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2002

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor, Dr. D. Yogi Goswami, for his patient guidance and support. I also owe a great deal of thanks to Dr. S. A. Sherif, Dr. Z. M. Zhang, Dr. J. Peterson and Dr. U. H. Kurzweg for their time and effort serving as my supervisory committee. Special thanks go to Dr. C. K. Hsieh for his invaluable assistance. My gratitude goes out to Mr. Charles Garreston whose marvelous experience and skills played a vital role in the design and construction of the experiment facility. In addition, I would also like to thank Feng Xu, Sanjay Vijayaraghavan, Gunnar Tamm, Viktoria Oberg Martin for their help and valuable advice. I thank Ms. Barbara Walker for her valuable assistance. Also, I feel honored to have worked with so many brilliant graduate students whose friendship and support make me feel at home when I am far away from my homeland.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS ii

NOMENCLATURE v

ABSTRACT ix

CHAPTERS

1 ENERGY RESOURCES 1

Geothermal Energy 2
Utilization of Geothermal Resources 3
Electricity generation 3
Direct heat uses 4
Environment impact 4
Solar Energy 5
Flat-Plate Collector 6
Concentrating Collector 7
Solar Pond 7

2 AMMONIA-BASED COMBINED POWER/COOLING CYCLE 9

Organic Rankine Cycle 9
Multi-Component Cycle 12
Ammonia-Based Combined Power/Cooling Cycle 15
Ammonia/Water Mixture as Working Fluid 20
Why Ammonia/Water? 20
Thermodynamic Properties of Ammonia/Water Mixture 22

3 SIMULATION AND PARAMETRIC ANALYSIS 23

Parametric Analysis 23
Irreversibility Analysis 44

4 OPTIMIZATION OF AMMONIA-BASED COMBINED POWER/COOLING CYCLE 55

Introduction to Optimization 55
Mathematical Formulation 55
Optimality Conditions 56
Unconstrained optimization 57

Constrained optimization 58
Generalized Reduced Gradient Algorithm 63
Description of the Problem 72
Variable Temperature Heat Source 72
Optimization Model for the Cycle 75
Optimization Program 79
Optimization Results 79
Optimization With Different Objective Functions 84
Effect of Ambient Temperature 87

5 APPLICATIONS OF THE NOVEL CYCLE 91

Solar Thermal Energy 91
Optimization Results 95
Effect of Water Storage Temperature 97
Waste Heat 98
Effect of Heat Source Temperature 100
Effect of Sink Temperature 103
Low Temperature Refrigeration 121

6 CONCLUSIONS 132

APPENDIX CYCLE SIMULATION PROGRAM WITH OPTIMIZATION 137

LIST OF REFERENCES 170

BIOGRAPHICAL SKETCH 174

NOMENCLATURE

COP_{ideal} : coefficient of performance for an ideal refrigeration cycle

$f_{2''}$: mass fraction at point 2", defined as $m_{2''}/m_1$

f_4 : mass fractions at point 4, defined as m_4/m_1

$f(\boldsymbol{x})$: objective function

g_R^T : generalized reduced gradient

$\boldsymbol{g}(\boldsymbol{x})$: inequality constraints

$\boldsymbol{h}(\boldsymbol{x})$: equality constraints

h_0 : enthalpy of the heat source fluid at ambient temperature

h_{ha}^{in} : inlet enthalpy of the heat source fluid

h_{ha}^{out} : outlet enthalpy of the heat source fluid

h_x : enthalpy of the working fluid at point x (refer to Fig. 2.7)

H : Hessian matrix

$\boldsymbol{L}$: lower bound of vector of free variables

L : Lagrange function

$\dot{m}_{ha}$: mass flow rate of heat source fluid

$\dot{m}_x$: mass flow rate of the working fluid at point x (refer to Fig. 2.7)

P_{high}: cycle high pressure

P_{low}: cycle low pressure

$\dot{Q}_{absorber}$: absorber heat rejection

$\dot{Q}_{boiler}$: boiler heat input

$\dot{Q}_{cool}$: refrigeration output

$\dot{Q}_{absorber}$: rectifier heat transfer

$\dot{Q}_{superheater}$: superheat input

s_0 : entropy of the heat source fluid at ambient temperature

s_{hs}^{in} : inlet entropy of the heat source fluid

s_{hs}^{out} : outlet entropy of the heat source fluid

T_{hs}^{in} : entrance temperature of heat source fluid

T_{hs}^{out} : exit temperature of heat source fluid

T_0 : ambient temperature

T_{absorber} : absorber temperature

T_{boiler} : boiler temperature

$T_{\text{rectifier}}$: rectifier temperature

$T_{\text{superheater}}$: superheater temperature

$T_{\text{boilermin}}$: minimum boiler temperature

$T_{\text{rectifiermin}}$: minimum rectifier temperature

T_x : temperature at state point x (refer to Fig. 2.7)

ΔT_{min} : minimum temperature difference required in the heat exchangers

ΔT_{pin} : temperature difference at pinch point in the boiler

$\Delta T_{pin}^{\text{min}}$: minimum temperature difference required at pinch point

U : upper bound of vector of free variables

w_{max}: availability or exergy per unit mass of heat source fluid

W_{net}: cycle net power output

W_p : pump work input

W_t : turbine work output

x : vector of free variables

x^* : local minimum

$x_{turbine}$: vapor quality at turbine exit

Greek:

ε : relative error

η_1: First law efficiency

η_2: Second law efficiency

λ : Lagrange multiplier

μ : Lagrange multiplier

Superscripts:

in : inlet condition

out : outlet condition

Subscripts:

0 : ambient condition

D: dependent variables

high : high pressure

hs : heat source

I : independent variables

ideal: ideal condition

low: low pressure

max : maximum

min: minimum

p : pump

pin : pin point

R: reduced gradient

t: turbine

x : state point x in Fig. 2.7

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

THERMODYNAMIC ANALYSIS AND OPTIMIZATION
OF A NEW AMMONIA BASED COMBINED POWER/COOLING CYCLE

By

Shaoguang Lu

May 2002

Chairman: D. Yogi Goswami
Major Department: Mechanical Engineering

A detailed thermodynamic analysis of a combined thermal power and cooling cycle is conducted. This cycle innovatively combines Rankine and absorption refrigeration cycles and uses ammonia-water mixture as a working fluid. It can provide power output as well as refrigeration with power generation as a primary goal. The concept of this cycle is based on the unique feature of a multi-component working fluid, varying temperature boiling process. Therefore, a better thermal match is obtained in the boiler between sensible heat source and working fluid. It also takes advantage of the low boiling temperature of ammonia vapor so that a temperature lower than ambient is achieved at the exit of the turbine. This cycle can be used as a bottoming cycle using waste heat from a topping cycle or as an independent cycle using low temperature sources such as geothermal and solar energy.

A parametric analysis has been conducted for the proposed cycle under idealized conditions. It helps to understand the behavior of the cycle and also shows that cycle working conditions could be optimized for best performance. The effect of irreversibilities on the cycle performance has also been studied.

An optimization algorithm, Generalized Reduced Gradient (GRG) algorithm, is introduced to optimize the performance of the proposed cycle. It searches a feasible region of free variables defined by their constraints to optimize the performance criteria. Second law efficiency is chosen as the primary optimization objective while the cycle could be optimized for any other performance parameter.

Cycle performance over a range of source and ambient temperatures was investigated. It was found that for a source temperature of 360K, which is in the range of flat plate solar collectors, both power and refrigeration outputs are achieved under optimum conditions. All performance parameters, including first and second law efficiencies, and power and refrigeration outputs decrease as the ambient temperature goes up. On the other hand, for a source of 440K, optimum conditions do not provide any refrigeration. However, refrigeration can be obtained even for this temperature under non-optimum performance conditions. In addition, some specific applications of the proposed cycle are studied.

CHAPTER 1
ENERGY RESOURCES

Energy is one of the building blocks of modern society. The growth of the modern society has been fueled by cheap, abundant energy resources. Today, 90% of our energy comes from fossil fuels. However, the resources of fossil fuels are limited and will deplete in the near future. Heavy reliance on fossil fuels since the beginning of the industrial revolution has also caused another problem, that is increased carbon dioxide concentration in the atmosphere, and probably increased global temperature. Warmer global temperatures can melt polar ice, leading to higher ocean levels and flooding of the cities near the seas. Acid rain caused by the emissions of coal-fired power plants harms trees and animals.

For all these reasons, it is urgent to develop and use energy resources that are clean and renewable, such as geothermal and solar energy. However, with the oil price being low and the environmental costs not accounted for, renewable energy sources, except perhaps wind energy, are still not cost competitive with fossil energy at present. It has been recognized that there is potential for reducing the costs by improving the performance of thermal power systems. Systems performance can be improved by employing new and innovative ideas in thermal power cycles (Goswami, 1998). In this dissertation, we make an investigation of a novel thermodynamic cycle, first suggested by Goswami (1995), which improves cycle efficiency and resource utilization by producing power and refrigeration in the same cycle. The new thermodynamic cycle can utilize low-

to-medium-temperature geothermal sources, and solar thermal sources with a high thermal efficiency, which may provide an opportunity to make them cost-competitive with fossil fuels.

Geothermal Energy

Geothermal energy is literally the heat contained in the Earth's interior. Geothermal resources come in five forms: hydrothermal fluids, geopressured brines, hot dry rock, magma, and ambient ground heat. Hydrothermal resources are reservoirs of steam or hot water, which are formed by water seeping into the earth and being heated by fractures or porous hot rock. Geopressured resources are deep buried waters at moderate temperatures that contain dissolved methane. Hot dry rock resources occur at depths of 5 to 10 miles beneath the earth's surface. Utilization of these resources involves injecting cold water down one well, circulating it through hot fractured rock, and drawing off the heated water from another well. Magma (or molten rock) resource has very high temperature, but no existing technology is able to use it. Ambient ground heat is the heat contained in soil and rocks at shallow depths.

Geothermal fluids were first exploited in the early nineteenth century in Larderello, Italy. After World War II, geothermal industry developed rapidly. The geothermal electric capacity reached 3433.086 MW_e in 1983 and 7173.5 MW_e in 1996. In 1995, the electrical energy generated was 38 billion kWh/year, representing 0.4% of the world total electrical energy which was 13,267 billion kWh in 1995 (Barbier, 1997). The total capacity of non-electric use of geothermal energy is estimated to be over 15,000 MW_t in 1997 (Mock et al., 1997).

The most common criterion for classifying geothermal resources is the enthalpy of the geothermal fluids. However, different experts have recommended different criteria. Table 1.1 lists some of them.

Table 1.1 Classification of geothermal resources (°C)

	(a)	(b)	(c)	(d)
Low enthalpy resources	<90	<125	<100	<=150
Intermediate enthalpy resources	90-150	125-225	100-200	-
High enthalpy resources	>150	>225	>200	>150

Sources:
(a) Muffler and Cataldi (1977).
(b) Hochstein (1990).
(c) Benderitter and Cormy (1990).
(d) Haenel et al. (1988).

Therefore, when we say a low, intermediate or high resource, we still have to indicate the temperature values or ranges.

If exploited correctly, geothermal energy can play an important role in the energy balance of many countries. Small geothermal resources, in particular, are capable of solving numerous local problems and of raising the living standards of small isolated communities.

Utilization of Geothermal Resources

Electricity generation is the most important form of utilization of high-temperature geothermal resources (>150°C). The low-to-medium-temperature resources (<150°C) are suited for many other applications.

Electricity generation

Electricity generation mainly takes place in conventional steam turbines and binary plants, depending on the characteristics of the geothermal resource. If hydrothermal fluids are totally or primarily steam, conventional steam turbines can be used. If

hydrothermal fluids are primarily water above 200° C, flash steam technology is used. The fluid is sprayed into a lower pressure zone, causing some of the fluid to flash into steam. The steam is then directed to a turbine. The Geysers, which is a dry steam power plant in northern California, is the world's largest single source of geothermal power. As for low-to-medium-temperature geothermal fluids, significant progress has been made in recent years due to the improvement of the binary fluid technology. A binary plant utilizes a secondary working fluid which has a low boiling point and high vapor pressure at low temperatures and thus is able to extract more heat from the heat source. By choosing appropriate secondary fluids, binary systems can utilize geothermal fluids in the temperature range of 85-170°C.

Direct heat uses

Direct uses of geothermal energy are traditional and well established world-wide. Space and district heating, agriculture applications and aquaculture are the best known and most widespread forms of utilization. Using geothermal energy for direct heat can not only help keep our environment clean, but also reduce fuel cost by as much as 80%, depending on the application and the industry (Dickson and Fanelli, 1995).

Environment Impact

Geothermal energy is considered as a clean energy, compared to fossil fuels. It has advantages of much fewer and more easily controlled atmospheric emissions, maintenance of ground water quality, and much smaller land requirements.

Geothermal power plants have an inherent advantage over fossil fuel plants because no combustion takes place. Geothermal plants emit no nitrogen oxides and very low amounts of sulfur dioxide, allowing them to easily meet the most strict clean air

standards. The steam at some steam plants contains hydrogen sulfide, but treatment processes remove more than 99.9% of those emissions. Typical emissions of hydrogen sulfide from geothermal plants are less than 1 part per billion. Geothermal plants emit 1/1000 to 1/2000 of the amount produced by fossil fuel plants (NREL, 1997).

To help eliminate waste water and subsidence phenomenon, spent geothermal fluid is treated and re-injected into the reservoir. Production and injection wells are lined with steel casing and cement to isolate the geothermal fluids from the environment. This technology not only disposes of the water where it came from, it also helps to maintain reservoir pressure and extend the longevity of the resource.

Geothermal installations require very little land use. In addition, other land uses may mingle with geothermal installations.

Solar Energy

The sun is the source of nearly all our energy (except radioactive sources and the tides). The energy of the sun comes from nuclear fusion in which four hydrogen protons are combined, releasing energy to form one helium atom. The earth receives energy from the sun at a rate of about 5000Q*/year, which is far greater than the present world energy consumption. Although the total quantity of solar energy incident upon the earth is immense, this energy is very diffuse. The amount of solar energy received per unit of area at the average sun-earth distance, measured outside the earth's atmosphere, is $1367 W/m^2$, which is known as the solar constant. It, therefore, needs systems that can gather and concentrate it, and convert it as efficiently as possible. Because of the earth's rotation and orbit around the sun, solar energy is cyclic both daily and seasonally. Solar

* $1Q \equiv 10^{21} J \equiv 3 \times 10^{14} kWh$

energy also suffers from atmospheric interference from clouds. Therefore, some efficient form of energy storage is necessary.

Solar energy is the only source from which we can use more energy than at present, without adding new thermal energy into the atmosphere. It may be used in many applications, such as in active and passive space heating and cooling, industrial process heating, desalination, water heating, and electricity generation. Solar energy may be converted to electricity by one of two means: solar thermal conversion and photovoltaic conversion. By thermal conversion, solar radiation is converted to heat that in turn is added to a working fluid going through thermodynamic cycle to produce mechanical work and electricity. For this system to be efficient and hence economical, it is necessary to collect and concentrate the diffuse solar radiation in an efficient manner to arrive at reasonably high temperature heat source. The solar collectors that may be used for thermal power are discussed below.

Flat-Plate Collector

A flat-plate collector is one of the most important types of solar collector because it is the simplest and has a wide range of important applications. A flat-plate collector is basically a black absorber surface with a means to transfer heat to a working fluid. A transparent cover and appropriate insulation around the sides and rear improve the efficiency of collection. The heat transfer fluid is generally water, but air may also be used. Other than simply providing hot water, one important application is the use of heat to operate a vapor-absorption refrigeration cycle for cooling applications. Flat-plate collectors can even be used to heat a fluid to operate thermodynamic cycles. The

advantage of the flat-plate collector is its low cost. Flat-plate collectors can provide temperature up to about 150°C (Winter, et al., 1991).

Concentrating Collector

Parabolic trough collectors use concentrators in the form of long troughs of parabolic cross sections, which are lined with reflecting surfaces to collect and concentrate the sun's radiation onto a focal linear conduit. The primary coolant flows through the conduit to carry the heat away. Because of their geometry, such troughs are usually designed to track the sun around only one axis. Parabolic trough collectors can provide temperatures up to about 400°C.

Parabolic dish collectors consist of a mirrored parabolic dish that tracks the sun and focuses the captured energy onto a receiver mounted at the focal point of the parabola. It can reach temperatures as high as 700°C, making it possible to use it for conventional thermal power plants.

Heliostat collectors consist of a number of heliostats and one central receiver. A heliostat is a reflecting mirror that is steerable so that it can reflect the sun's rays onto the central receiver at almost all times during the daylight hours. In essence, they keep the sun stationary as far as the receiver is concerned. These collectors can achieve temperatures of up to about 1500°C. They are suitable for large-scale power plants.

Solar Pond

A solar pond combines both the collection and the thermal storage subsystems. The concept of the solar pond is based upon the observation by Dr. Kalecsinski that natural saline lakes can have a steep temperature gradient with depth. This was found to

be associated with a concentration gradient in the dissolved solids, resulting in a density gradient sufficient to suppress thermal convection.

A typical solar pond is 1-2m deep and covers several hundreds to thousands of hectares. A solar pond is divided into three layers. The central layer is nonconvective and isolates a low-salinity (and density) convective layer at the top and high-salinity (and density) convective layer at the bottom. In a thermodynamic cycle, the warm water at the bottom is used as the heat source, which is insulated from the atmosphere above. The cool water at the top is used as the heat sink. Because the temperature difference between the heat source and heat sink is not great enough, the working fluid is usually an organic fluid. The temperature a solar pond can reach at the bottom is around 80-90°C. Besides electric generation, solar ponds may be used for industrial applications such as space heating and cooling, crop drying, desalination and other process heat.

A barrier to wide use of solar energy systems has been the economic viability while environmental and social costs are not included. Consideration of these costs can make the viability of solar energy much more favorable. In principle, a solar energy system should seek to provide an optimal combination of efficient performance, low initial and running costs, robustness and durability.

CHAPTER 2
AMMONIA-BASED COMBINED POWER/COOLING CYCLE

Compared to fossil fuels, heat sources, such as geothermal energy, solar energy and waste heat, are typically low temperature. Conventional steam Rankine cycle, which works well for the utilization of fossil fuels, has very low efficiency for low temperature heat sources. For such low temperature heat sources, new concepts need to be developed to effectively utilize them. Some research and development projects have resulted in better efficiency for power conversion from low temperature heat sources. Organic Rankine cycles have been used in geothermal electric industry for more than a decade. The Kalina cycle has also been studied. In this dissertation, a novel cycle suitable for low temperature sources will be studied in detail.

Organic Rankine Cycle

The use of an organic fluid in the Rankine cycle is an innovative way to improve the performance of a thermal power plant which utilizes low-to-medium-temperature resources. The organic fluids have low boiling temperature. For example, isobutane (C_4H_{10}) has a boiling temperature of –11.73 ^{o}C at atmospheric pressure. This feature makes organic fluids suitable for low temperature heat sources. The principle and configuration of an organic Rankine cycle are the same as for the steam Rankine cycle only the working fluid is replaced by an organic fluid. The cycle shown in Fig. 2.1 uses isobutane as the secondary working fluid (Phair, 1994). Heat from geothermal brine vaporizes the isobuatne in the brine heat exchanger. The spent brine is re-injected back to

the ground, and isobutane vapor drives a turbine generator. The turbine exhaust is delivered to an air-cooled condenser, where the vapor is condensed. Liquid isobutane drains to an accumulator and is then pumped back to the brine heat exchangers to repeat the cycle.

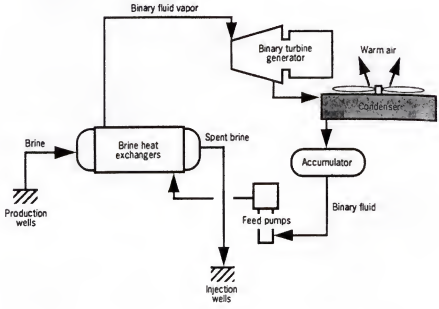


Figure 2.1 A Binary Organic Rankine Cycle for Geothermal Energy

Despite the fact that the use of an organic Rankine cycle improves the performance of a geothermal power plant, its first law thermal efficiency is not much different from that of a steam Rankine cycle for the same thermal boundaries. In fact, the first law thermal efficiency is often slightly lower for an organic Rankine cycle. The advantage comes from the fact that organic working fluid is able to extract more heat from the geothermal heat source than water and thus the cycle has a higher second law thermal efficiency. In a Rankine cycle, the vaporization process is isothermal. It is a horizontal line on a T-S diagram (Fig. 2.2). This characteristic results in the poor match

between a sensible heat source and a pure working fluid due to the limitation of pinch point in the heat exchanger. Although the use of an organic fluid instead of water will not eliminate this disadvantage, it does shorten the vaporization process. Organic fluids have a low ratio of heat of vaporization to specific heat capacity. For example, for isobutane at 100°C, its specific heat capacity is 3.366 kJ/kg.K and the latent heat of vaporization is 210.94 kJ/kg, which gives the ratio as 62.7. However, for water at 100°C, this ratio is 535.1. The lower ratio of the heat of vaporization to specific heat capacity gives organic fluids the ability to take more heat out of heat sources in the preheater.

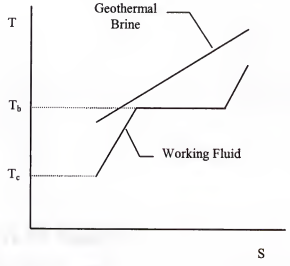


Figure 2.2 Temperature-Entropy Diagram for Heat Exchange

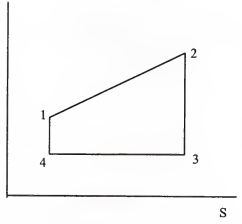
In addition, an organic Rankine cycle has other advantages:

- For an organic fluid, complete expansion may take place outside the saturation curve.
- The enthalpy drop is small in the turbine and it is possible to design a single-stage turbine with high efficiency.
- Although the mass flow rate of organic fluid is higher, its high density at the exhaust makes the volumetric flow rate about sixteen times lower and therefore the size of turbine is considerably smaller.

Due to these apparent advantages, the organic Rankine cycle is widely used today to generate electricity from low-to-medium-temperature geothermal resources.

Multi-Component Cycle

Though the use of an organic fluid in a Rankine cycle makes it possible to extract more heat from geothermal heat sources, there still exists a big temperature gap between the heat source and the working fluid. From Fig. 2.2, we find that if the heat gain process of the working fluid is a parallel line with the heat rejection line of the heat source, it will have the best thermal match between them. In light of this recognition, Lorenz (1894) invented the Lorenz cycle, which is the ideal power cycle for a sensible heat source. This cycle has a triangular shape on a temperature and entropy diagram (Fig. 2.3), generating the least entropy during the heat transfer process (Kalina, 1984). The least production of entropy, in turn, yields the highest thermodynamic efficiency. The Lorenz cycle is composed of four processes:



1-2 Heat gain at variable temperatures

2-3 Isentropic expansion

3-4 Isothermal heat rejection

4-1 Isentropic compression

Figure 2.3 Temperature-Entropy Diagram of Lorenz Cycle

There are several ways to approach the Lorenz cycle. One is the use of a multi-pressure boiler, which is widely accepted in the industry. Compared to a single-pressure

boiler, a multi-pressure boiler has a better thermal match. However, the improvement is only moderate unless the number of boiler steps is very large. Due to the technical and economic consideration, the number of boiler steps usually will not be more than three. The use of a supercritical cycle (Fig. 2.4), especially with organic working fluids, can theoretically achieve a triangular shape of the cycle, but requires extremely high pressure in the boiler, which in turn adversely affects the turbine performance (Kalina 1984). Milora and Tester (1976) have given a detailed discussion of the supercritical cycle.

The third alternative way is to use a multi-component working fluid, which has been intensively studied during the past two decades. A multi-component working fluid boils at variable temperatures with a change in the liquid composition of the components. This variable temperature boiling process yields a better thermal match with the sensible heat source than the constant temperature boiling process, and is close to a triangular shape.

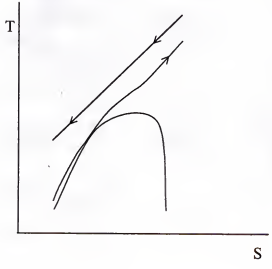


Figure 2.4 Temperature-Entropy Diagram of a Supercritical Cycle

Two important kinds of multi-component fluids, mixed organic working fluids and ammonia/water mixtures, have been proposed in the literature. One thermodynamic cycle using ammonia/water mixture is the Kalina cycle, which was introduced by Kalina (1983). A comparison of the Kalina cycle to the Rankine cycle by El-Sayed and Tribus shows a 10% to 20% improvement in thermal efficiency (El-Sayed and Tribus, 1985). Although Kalina is recognized for introducing the ammonia-water working fluids into power cycles (Kalina, 1984; Kalina et al., 1986; Kalina and Leibowitz, 1987; Kalina and Tribus, 1990), Maloney and Robertson (1953) studied an absorption-type power cycle using a mixture of ammonia and water as the working fluid in the early fifties. Marston (1990a, 1990b), Park and Sonntag (1990), and Ibrahim and Klein (1996) also analyzed the Kalina cycle. Their studies showed the advantages of the Kalina cycle over the conventional Rankine cycle under certain conditions. The first Kalina combined-cycle plant has been tested (Leibowitz and Mirolli, 1997). Rogdakis and Antonopoulos (1991) proposed a triple stage power cycle, which replaces the distillation condensation of the Kalina cycle with the absorption condensation process. The absorption condensation process eliminates the need to use a large number of heat exchangers and simplifies the ammonia-water power cycle.

Since the cycle proposed by Rogdakis and Antonopoulos still uses ammonia-water vapor flowing through the turbine, the exit temperature must be relatively high in order to avoid condensation in the turbine. In allusion to this limitation, Goswami (1995, 1998) proposed a new power cycle, which is called ammonia-based combined power/cooling cycle. While retaining the advantages of the Kalina cycle and the Rogdakis and Antonopoulos cycle as identified above, it removes their constraints by

using very high concentration ammonia vapor in the turbine, which allows the vapor to expand to a much lower temperature without condensation. This cycle will be able to use low quality heat sources, including geothermal energy and solar energy, while achieving high thermal efficiency.

Ammonia-Based Combined Power/Cooling Cycle

Goswami (1995, 1998) proposed a new thermodynamic cycle that improves the cycle efficiency and therefore resource utilization by producing power and refrigeration in the same cycle. The proposed new cycle uses ammonia/water as the working fluid in an innovative combination of two thermodynamic cycles, the ammonia based Rankine cycle and the ammonia-absorption refrigeration cycle.

Figure 2.5 shows a schematic diagram of this ammonia-based combined power/cooling cycle. Ammonia-water solution is pumped to a high pressure (state 2) after leaving the absorber. It passes through a heat exchanger to recover the heat carried by the weak solution (state 3). In the boiler, ammonia vapor is boiled off and separated from the weak solution (state 4). After expanding through a turbine to generate power, ammonia reaches a low temperature (state 5) and provides cooling in the cooler (state 6). In the absorber, ammonia vapor is absorbed by the weak solution from the boiler to form the basic ammonia/water liquid solution again (state 1) to repeat the cycle. The heat carried by the ammonia-water weak solution leaving the boiler (state 7) is recovered in the heat exchanger (state 8). After passing through a pressure-reducing valve (state 9), the weak solution returns to the absorber.

One significant contribution of this novel cycle is that it essentially removes the limitation of the heat sink temperature of the environment in the Rankine cycle part. The

fluid can expand to a temperature much lower than the environment. Thereby it can provide power output as well as refrigeration.

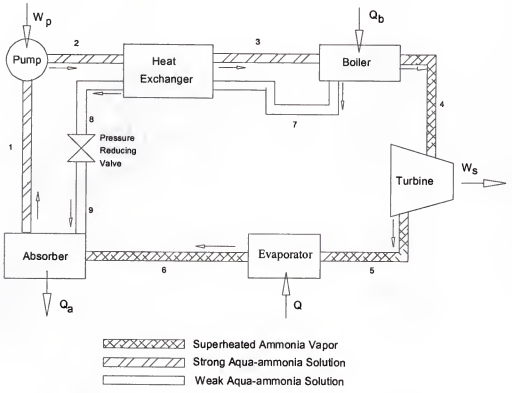


Figure 2.5 Conceptual Ammonia-Based Combined Power/Cooling Cycle

This ammonia-based combined power/cooling cycle finds a creative way to realize the triangular shape of the Lorenz cycle. Its concept is based on the unique feature of a multi-component working fluid, in this case, ammonia/water. The boiling temperature of ammonia/water increases as the boiling process proceeds; thus a better thermal match is obtained in the boiler between the sensible heat source and the working fluid. Consequently, the cycle has a better thermodynamic performance. As conceptually shown in Fig. 2.6, this ammonia-based combined power/cooling cycle provides extra work (shade area) over the conventional Rankine cycle. It also takes advantage of the low

boiling temperature of ammonia. Ammonia vapor can expand to a temperature lower than ambient in the turbine while it is still in vapor or high quality two-phase state. The cycle can effectively utilize a heat source with a temperature lower than 100°C, therefore making it very promising for geothermal resources, low-cost solar thermal collectors and waste heat.

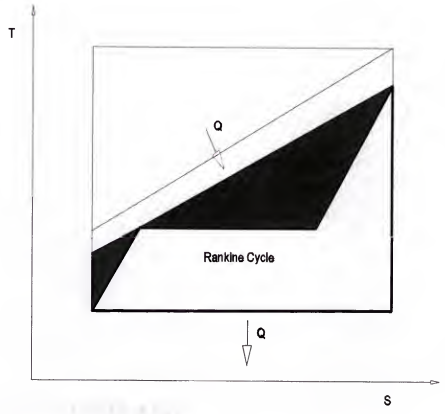


Figure 2.6 T-S diagram showing advantage of the proposed cycle over a conventional Rankine cycle

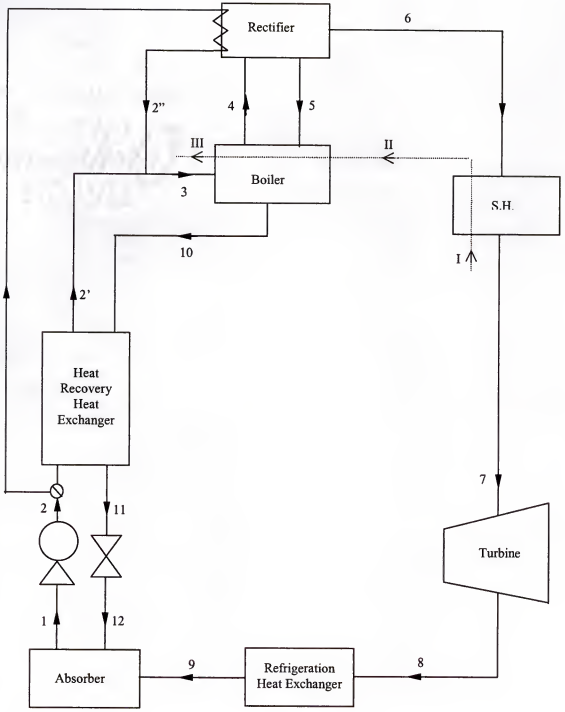


Figure 2.7 A practically designed ammonia-based combined power/cooling cycle

While the conceptual ammonia-based combined power/cooling cycle has been described as above, there is one thing still left to be addressed. To expand to a lower temperature than ambient in the turbine, ammonia vapor must have high purity. Otherwise, the moisture within the vapor would condense and damage the turbine blades. Usually, ammonia vapor generated by a boiler is not pure enough to maintain high quality at very low temperatures. Therefore, ammonia vapor has to be purified before entering the turbine. In the practically designed ammonia-based combined power/cooling cycle shown in the Figure 2.7, a rectifier is placed on top of the boiler to purify the ammonia vapor. High concentrated vapor is then superheated in the superheater. A part of the ammonia strong solution is diverted to provide cooling in the rectifier and recuperate the heat.

Goswami and Xu (1999) analyzed a simple case of the combined power/cooling cycle for a 400 K constant temperature heat source in the boiler, a 410K constant temperature heat source in the superheater, and an ambient temperature of 280K. The cycle was evaluated assuming ideal processes; that is, the irreversibilities associated with the heat transfer processes, the pump and the turbine were neglected. Its state-point conditions are listed in table 2.1.

Table 2.1 shows a great advantage of this ammonia-based combined power/cooling cycle. A first law efficiency of 23.54 percent is achieved. By contrast, the Carnot cycle efficiency operating between 410K and 280K is 31.7%. The cycle efficiency of a conventional steam Rankine cycle between the same source and sink temperatures will be much lower.

Table 2.1 Typical Operating Conditions

Point	T(K)	P (bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-214.1	-0.1060	0.5300	1.0000
2	280.0	30.0	-211.4	-0.1083	0.5300	1.0000
3	378.1	30.0	246.3	1.2907	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	205.8	1.1185	0.6763	0.0366
6	360.0	30.0	1373.2	4.1520	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	257.0	2.0	1148.9	4.5558	0.9921	0.1997
9	280.0	2.0	1278.7	5.0461	0.9921	0.1997
10	400.0	30.0	348.2	1.5544	0.4147	0.8003
11	300.0	30.0	-119.0	0.2125	0.4147	0.8003
12	300.0	2.0	-104.5	0.2718	0.4147	0.8003

Boiler heat input:	390.4 kJ/s
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	385.8 kJ/s
Refrigeration output:	25.9 kW
Turbine work output:	76.0 kW
Turbine liquid fraction:	0.0692
Turbine vapor fraction:	0.9308
Pump work input:	2.7 kW
Total heat input:	421.6 kJ/s
Net power and refrigeration output:	99.23 kW
Thermal efficiency:	23.54 %
Carnot efficiency (between 410K and 280K):	31.7 %

Ammonia/Water Mixture as Working Fluid

Why Ammonia/Water?

As a multi-component working fluid, ammonia/water solution boils at variable temperatures, which makes it a good choice for applications of sensible heat sources. This feature is clearly illustrated in Fig. 2.8 (The pressure in the figure is atmospheric pressure). Ammonia has lower boiling temperature than water under the same pressure, so it has stronger tendency to escape from the solution. When ammonia starts to boil off

from the solution, its concentration in the solution goes down. Consequently, the bubble temperature of the ammonia/water solution goes up.

The slope of the boiling process curve of the ammonia/water provides an excellent match to the energy-releasing curve of a sensible heat source. The temperature gap between the heat source and the working fluid is reduced. Therefore, less available energy is lost during the heat transfer process. As a result, the overall performance of the power cycle is improved.

Ammonia/water solution has a low boiling temperature (See Fig. 2.8). For 50% ammonia 50% water mixture, the atmospheric boiling point is –4°C. This feature allows ammonia/water to be used for low temperature heat source. In the same way, almost pure ammonia vapor has a very low dew point, which allows it to be expanded to a very low temperature in the turbine without causing too much condensation.

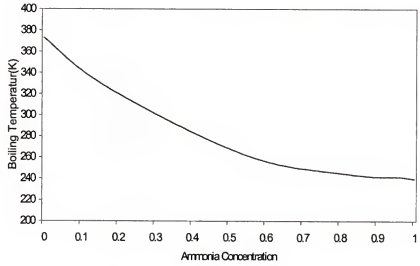


Figure 2.8 Boiling Temperature Diagram for Ammonia-Water Solutions

The molecular weight of ammonia is very close to steam. This similarity makes it possible to utilize the existing turbine technology in an ammonia power cycle.

Thermodynamic Properties of Ammonia/Water Mixture

Ammonia/water mixture has been used in absorption refrigeration cycles for several decades. However, the temperature and pressure range of ammonia/water in the power cycle is much higher than that in an absorption refrigeration cycle. There is great demand to extend thermodynamic data of ammonia/water mixture to higher temperature and pressure.

Many studies have been done on the thermodynamic properties of ammonia/water mixtures. Xu and Goswami (1999) presented a method that combines the Gibbs free energy method for mixture properties and bubble and dew point temperature equations for phase equilibrium.

This method combines the advantages of the two and avoids the need for iterations for phase equilibrium by the fugacity method. Xu (1996) developed a computer program for ammonia-water thermodynamic properties. Xu and Goswami (1999) compared the computational results with accepted experimental data in the literature which showed very good agreement.

CHAPTER 3
SIMULATION AND PARAMETRIC ANALYSIS

The ammonia-based combined power/cooling cycle and one of its typical working conditions are described in detail in chapter 2. In this chapter, the performance of the cycle is analyzed thermodynamically.

Parametric Analysis

Parametric analysis is the first step to analyze a thermodynamic cycle. By varying one parameter at one time, it is used to study the effect of different parameters on the cycle performance. It helps us to understand the thermodynamic cycle and estimate good working conditions. In this section, a parametric analysis of this ammonia-based combined power/cooling cycle is presented to evaluate its performance. The analysis assumes idealized conditions, neglecting the irreversibilities associated with the cycle. The parameters being studied include turbine inlet pressure, boiler temperature, rectifier temperature, superheater temperature, absorber temperature and pressure.

In the parametric analysis, ammonia-water properties are needed at different state points. The property program developed by Xu (1996) is used in the analysis. This program uses Gibbs free energy equations for pure ammonia and water properties, and bubble and dew point temperature equations for vapor-liquid equilibrium and has shown good agreement with the experimental data (Xu and Goswami, 1999). However, some modifications have been made to Xu's program in order to increase the speed and improve its accuracy. In the property program, iterations are needed in several functions.

For example, the available formulae only calculate entropy or enthalpy from pressure, temperature, and concentration, not vice versa. Therefore, if we need to calculate the temperature of a state point when we know the pressure, concentration and entropy, trial and error method is used. Xu's program starts with an initial temperature guess, which is then increased in small increments until the solution is found. In the present program, a bisection method is used. By using the bisection method, it saves computation time and improves accuracy, which is very crucial to the later optimization work.

The Basic Equations used in the simulation program are as follows.

Boiler heat input:

$$\dot{Q}_{boiler} = \dot{m}_4 h_4 + \dot{m}_{10} h_{10} - \dot{m}_3 h_3 - \dot{m}_5 h_5 \tag{3.1}$$

Superheat input:

$$\dot{Q}_{superheater} = \dot{m}_6 (h_7 - h_6) \tag{3.2}$$

Rectifier heat transfer:

$$\dot{Q}_{rectifier} = \dot{m}_5 h_5 + \dot{m}_6 h_6 + \dot{m}_{2'} h_{2'} - \dot{m}_4 h_4 - \dot{m}_{2'} h_2 \tag{3.3}$$

Absorber heat rejection:

$$\dot{Q}_{absorber} = \dot{m}_1 h_1 - \dot{m}_{12} h_{12} - \dot{m}_9 h_9 \tag{3.4}$$

Pump work input:

$$W_p = \dot{m}_1 (h_2 - h_1) \tag{3.5}$$

Turbine work output:

$$W_t = \dot{m}_7 (h_7 - h_8) \tag{3.6}$$

Refrigeration output:

$$\dot{Q}_{cool} = \dot{m}_8 (h_9 - h_8) \tag{3.7}$$

First law thermal efficiency:

$$\eta_1 = \frac{W_t - W_p + \dot{Q}_{cool}}{\dot{Q}_{boiler} + \dot{Q}_{superheater}} \tag{3.8}$$

The temperature of the working fluid at point 2" is set as 5K below rectifier temperature. Its flow rate is calculated as:

$$\dot{m}_{2"} = \frac{\dot{Q}_{rectifier}}{(h_{2"} - h_2)} \tag{3.9}$$

Where $\dot{m}_x$ is the mass flow rate of the working fluid at point x, and h_x is the enthalpy of the working fluid at point x (refer to Fig. 2.7).

In the following paragraphs, the effect of different parameters on the cycle performance is scrutinized.

Effect of Turbine Inlet Pressure

Turbine inlet pressure was varied between 18 - 32 bars and the effect of such variation on the cycle thermal efficiency, vapor fraction at the exit of the boiler/rectifier, turbine work output, and cooling capacity is shown in Figures 3.1 - 3.4.

Fig. 3.1 shows that vapor fraction goes down almost linearly with turbine inlet pressure. Fig. 3.2 shows that turbine work output also goes down with the turbine inlet pressure, but the curve is not straight. The turbine work output is the product of vapor flow rate and enthalpy drop in the turbine. When vapor fraction goes down, turbine work output goes down correspondingly. However, enthalpy drop across the turbine increases as the turbine inlet pressure increases. The gain eases the loss caused by the decrease of the vapor fraction, thus making the drop of turbine work output slower than that of vapor fraction.

Although turbine work output decreases as the turbine inlet pressure increases, the thermal efficiency goes up first to a maximum and then goes down. This result is shown in Fig. 3.3. Similar trend is shown in Fig. 3.4 for cooling capacity.

In addition, when the concentration of the ammonia strong solution increases, it leads to an increase of thermal efficiency, vapor fraction, turbine work output and cooling capacity. However, the concentration of the ammonia strong solution can not exceed its saturation limit at a given absorber pressure and temperature. At 2 bar and 280 K, the saturation concentration of the ammonia solution is slightly above 0.53.

Effect of Boiler Temperature

Boiler temperature was varied between 390 - 420 K and the effect of such variation on the cycle thermal efficiency, vapor fraction at the exit of the boiler/rectifier, turbine work output and cooling capacity is shown in Figures 3.5 – 3.8.

Fig. 3.5 shows that vapor fraction goes up almost linearly as the boiler temperature goes up. Consequently, the turbine work output and cooling capacity also increase (Figures 3.6 and 3.7). However, since the heat input increases as the boiler temperature increases, the thermal efficiency will increase to a limit even though turbine work output and cooling capacity keep increasing (Fig. 3.8).

Effect of Rectifier Temperature

Rectifier temperature was varied between 350 - 400 K and the effect of such variation on the cycle thermal efficiency, vapor fraction at the exit of the boiler/rectifier, turbine work output and cooling capacity is shown in Figures 3.9 – 3.12.

Fig. 3.9 shows that vapor fraction decreases as the rectifier temperature goes down from 400K to 350K. As a result, turbine work output decreases (Fig. 3.10).

However, cooling capacity increases as the rectifier temperature goes down as shown in Fig. 3.11. When the rectifier temperature goes down, there is less moisture in the ammonia vapor, which allows it to expand to a lower temperature. From Fig. 3.11, we can see that no cooling is available when rectifier temperature is higher than 385 K. Fig. 3.12 shows the change in cycle thermal efficiency with the rectifier temperature, which is the combined effect of the results shown in Fig. 3.10 and 3.11. The thermal efficiency decreases first as the rectifier temperature increases, but goes up again when the rectifier temperature is greater than 385 K. Below 385 K, when the rectifier temperature increases, the cooling capacity drops faster than the increase of the turbine work output. Thermal efficiency thus decreases. Above 385 K, higher rectifier temperature helps increase turbine work output, therefore the thermal efficiency goes up again.

Effect of Superheater Temperature

Superheater temperature was varied between 400 - 500 K and the effect of such variation on the cycle thermal efficiency, vapor fraction at the exit of the boiler/rectifier, turbine work output and cooling capacity is shown in Figures 3.13 – 3.16.

As expected, vapor fraction is not affected by superheat temperature (Fig. 3.13). Figures 3.14 and 3.15 show that a higher superheater temperature benefits the turbine work output but hurts the cooling capacity which eventually disappears. Because cooling capacity drops faster than the increase in turbine work output, thermal efficiency drops steadily as the superheat temperature increases (Fig. 3.16). This trend is only reversed after cooling capacity approaches zero.

Effect of Absorber Temperature

Absorber temperature was varied between 280 - 310 K and the effect of such variation on the cycle thermal efficiency, vapor fraction at the exit of the boiler/rectifier, turbine work output and cooling capacity is shown in Figures 3.17 – 3.20.

From Figures 3.17 to 3.20, one can see that the absorber temperature has much bigger effect on the cycle performance than the previous parameters. With less than 30 K absorber temperature increase, the thermal efficiency of the cycle using saturated ammonia solution drops sharply from about 22% to 0, assuming that the boiler pressure is kept at 25 bars. The reason is that absorber temperature has a great effect on the concentration of the ammonia strong solution. When the absorber temperature goes up, the concentration falls very fast. Consequently, the amount of ammonia vapor generated in the boiler falls drastically as shown in Fig. 3.18. Same trends can be seen for turbine work output and cooling capacity in Figures 3.19 and 3.20.

However, if the boiler pressure is lower than 25 bars, thermal efficiency, vapor fraction, turbine work output and cooling capacity of the cycle using saturated ammonia solution can be nonzero at 307 K absorber temperature (shown in Figures 3.21 – 3.24).

Effect of Absorber Pressure

Absorber pressure was varied between 1 - 3 bar to see the effect of such variation on the cycle thermal efficiency, vapor fraction at the exit of the boiler/rectifier, turbine work output and cooling capacity. Figures 3.25 to 3.28 show the results.

Fig. 3.25 shows that vapor fraction increases with the absorber pressure. Since high absorber pressure makes high concentration of saturated ammonia solution in the absorber, more ammonia vapor can be produced in the boiler. Consequently, turbine

work output increases (Fig. 3.26). However, cooling capacity does not follow the same trend (Fig. 3.27). It increases first and then decreases. There is another factor affecting the cooling capacity. While the increase of the absorber pressure leads to a higher concentration for ammonia strong solution, it also raises the back pressure of the turbine. Therefore, cooling capacity falls after certain pressure. Same trend is seen for the thermal efficiency in Fig. 3.28.

In the above analysis of the effect of the turbine inlet pressure, boiler temperature, rectifier temperature and superheater temperature, three different concentration values of the ammonia strong solution are used. But in the analysis of the last two parameters, absorber temperature and pressure, fixed concentration values cannot be used any more. The saturation concentration of the ammonia strong solution depends on the absorber temperature and pressure. When the absorber temperature or pressure changes, the saturation concentration of the ammonia strong solution changes correspondingly. So instead of using the earlier fixed values of concentrations, 90%, 95% and 100% of the saturation concentration values for the chosen pressure and temperature conditions are used to study the effect of the concentration on the cycle performance.

Parametric analysis only varies one parameter at a time while keeping other parameters at fixed values. Although an optimal value for the varying parameter could be found for the set of fixed values of the remaining parameters, the value would be different for other sets of fixed values. To find the true optimum working condition, an optimization algorithm, which varies all the free parameters at the same time to search for the best objective, is needed. In the next chapter, such an optimization algorithm is presented and used for the optimization of the combined power/cooling cycle.

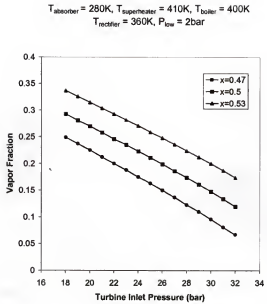


Figure 3.1 Effect of Turbine Inlet Pressure on Vapor Fraction

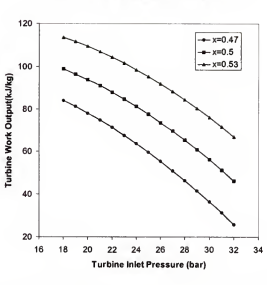


Figure 3.2 Effect of Turbine Inlet Pressure on Turbine Work Output

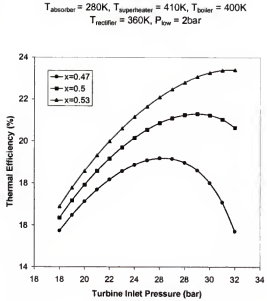


Figure 3.3 Effect of Turbine Inlet Pressure on Thermal Efficiency

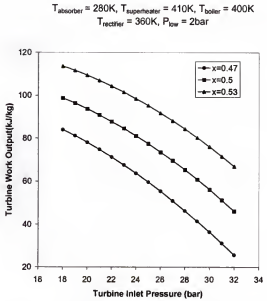


Figure 3.4 Effect of Turbine Inlet Pressure on Cooling Capacity

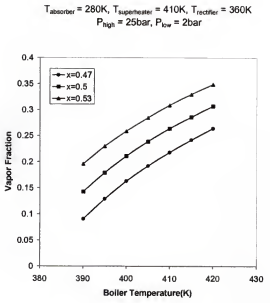


Figure 3.5 Effect of Boiler Temperature on Vapor Fraction

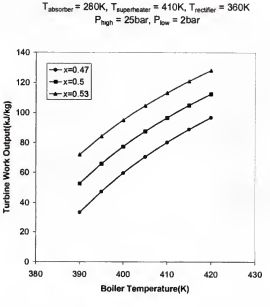


Figure 3.6 Effect of Boiler Temperature on Turbine Work Output

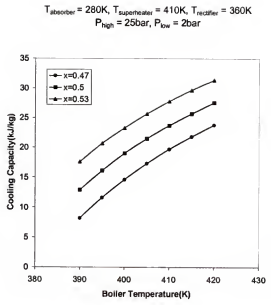


Figure 3.7 Effect of Boiler Temperature on Cooling Capacity

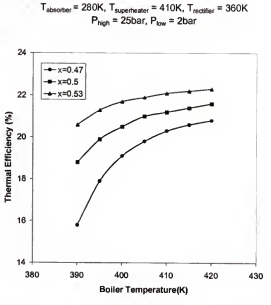


Figure 3.8 Effect of Boiler Temperature on Thermal Efficiency

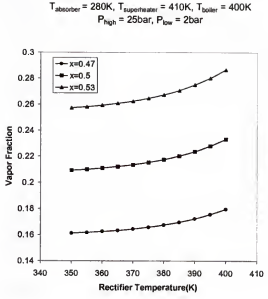


Figure 3.9 Effect of Rectifier Temperature on Vapor Fraction

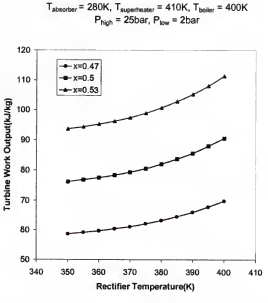


Figure 3.10 Effect of Rectifier Temperature on Turbine Work Output

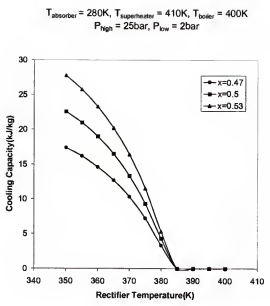


Figure 3.11 Effect of Rectifier Temperature on Cooling Capacity

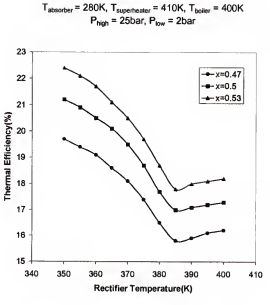


Figure 3.12 Effect of Rectifier Temperature on Thermal Efficiency

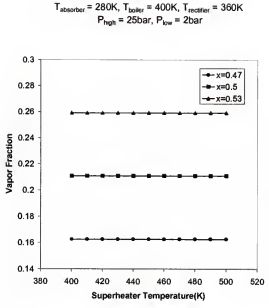


Figure 3.13 Effect of Superheater Temperature on Vapor Fraction

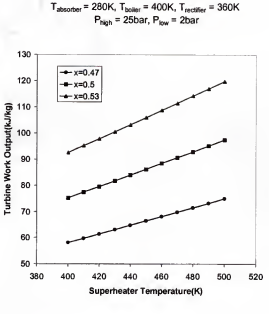


Figure 3.14 Effect of Superheater Temperature on Turbine Work Output

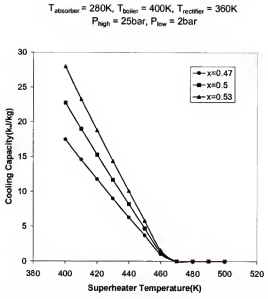


Figure 3.15 Effect of Superheater Temperature on Cooling Capacity

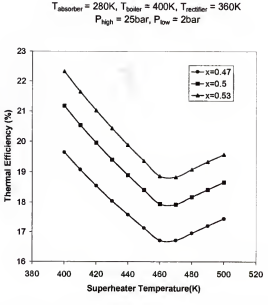


Figure 3.16 Effect of Superheater Temperature on Thermal Efficiency

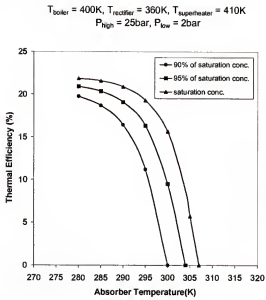


Figure 3.17 Effect of Absorber Temperature on Thermal Efficiency

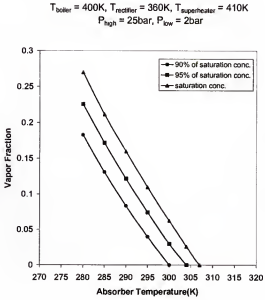


Figure 3.18 Effect of Absorber Temperature on Vapor Fraction

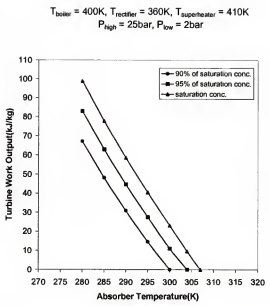


Figure 3.19 Effect of Absorber Temperature on Turbine Work Output

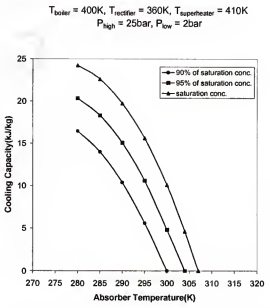


Figure 3.20 Effect of Absorber Temperature on Cooling Capacity

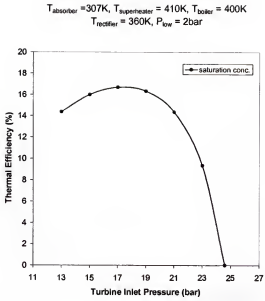


Figure 3.21 Thermal Efficiencies at Different Turbine Inlet Pressures

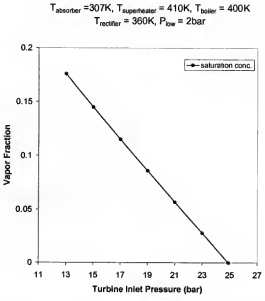


Figure 3.22 Vapor Fractions at Different Turbine Inlet Pressures

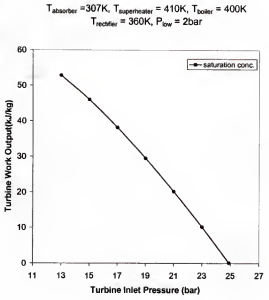


Figure 3.23 Turbine Work Outputs at Different Turbine Inlet Pressures

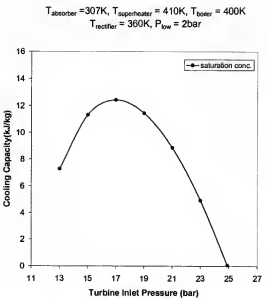


Figure 3.24 Cooling Capacities at Different Turbine Inlet Pressures

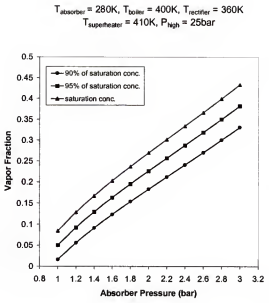


Figure 3.25 Effect of Absorber Pressure on Vapor Fraction

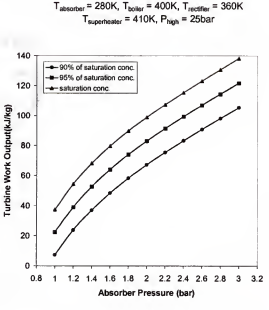


Figure 3.26 Effect of Absorber Pressure on Turbine Work Output

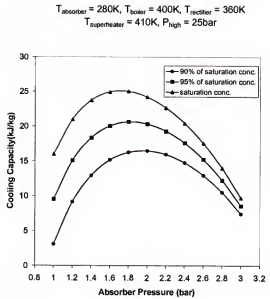


Figure 3.27 Effect of Absorber Pressure on Cooling Capacity

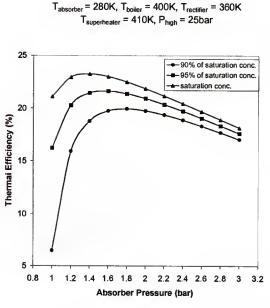


Figure 3.28 Effect of Absorber Pressure on Thermal Efficiency

Irreversibility Analysis

In the parametric analysis, we assumed ideal processes for the cycle. In real life, there are irreversibilities associated with every part of the cycle. The pump and turbine have less than isentropic efficiencies. Components have pressure drops. These irreversibilities will have negative effects on the performance of the cycle. In this section, the effects of the irreversibilities associated with the cycle are analyzed quantitatively. First, they are studied individually so that we know how much each factor affects the cycle performance. Then, their combined effect on the cycle is studied. The typical working condition presented in table 2.1 is used as a reference. That typical working condition is obtained by assuming the following.

Boiler Exit Temperature: 400K
Boiler Exit Pressure : 30bar
Turbine Inlet Temperature: 410K
Rectifier Temperature: 360K
Absorber Temperature: 280K
Absorber Pressure: 2 bar
Basic Solution Concentration: 0.53

The same assumptions are used in the irreversibility analysis so comparison could be easily made.

In table 2.1, the processes in the pump and the pressure-reducing valve are simplified as constant temperature processes while they are actually constant entropy and enthalpy processes respectively under ideal conditions. Although the assumption causes little effect on the results, the new results, considering these processes as constant entropy and enthalpy processes respectively, are presented in table 3.1.

In table 3.2, the turbine efficiency is assumed as 90% (Drbal, et. al., 1996). By comparing tables 3.2 and 3.1, it can be seen that turbine inefficiency has significant effect on the overall cycle efficiency. The first law efficiency drops from 23.27% to 18.52%. Due to the irreversibility in the turbine, the exhaust temperature of the turbine is higher even though the pressure ratio is the same. Less energy is converted into mechanical work in the turbine, and the turbine work output drops from 76.1 kW to 68.5 kW per kg of basic solution. At the same time, higher turbine exhaust temperature means less cooling capacity. Cooling capacity drops from 26.0 kW to 18.4 kW per kg of basic solution.

In table 3.3, the pump efficiency is assumed as 80% (Drbal, et. al., 1996). The effect of pump inefficiency on the cycle is insignificant. Pump work input increases from 3.4 kW to 4.2 kW per kg of basic solution, causing cycle efficiency to drop slightly.

In table 3.4, a 5% pressure loss was assumed in the boiler (Bhatt, et. al., 1994). The result shows it has almost no negative effect on the cycle performance. Only slight more pump work is required to boost the boiler inlet pressure to compensate for the pressure loss in the boiler.

In table 3.5, a 5% pressure loss was assumed for the superheater (Bhatt, et. al., 1994). The result shows that this pressure loss has some negative effect on the cycle performance. The first law efficiency drops from 23.27% to 22.64%. Due to the pressure loss in the superheater, turbine inlet pressure drops. Therefore, turbine work output drops and the exhaust temperature increases. Higher exhaust temperature causes less cooling capacity. Comparing tables 3.5 and 3.1, it can be seen that turbine work output drops

from 76.1 kW to 75.2 kW per kg of basic solution, and cooling capacity drops from 26.0 kW to 24.3 kW per kg of basic solution.

In table 3.6, a 5% pressure loss is assumed for both streams in the solution heat exchanger (Bhatt, et. al., 1994). The result shows that the pressure loss in the solution heat exchanger does almost no harm to the cycle performance. First law efficiency is essentially not affected. Only a little bit extra pump work is needed.

Table 3.7 shows that a 5% pressure loss in the cooler causes the cycle efficiency to drop from 23.27% to 22.73%. Due to the pressure loss in the cooler, turbine exhaust pressure is raised. As a result, the pressure ratio of the turbine decreases. Turbine work output decreases and exhaust temperature increases. Therefore, cooling capacity also decreases. Comparing tables 3.7 and 3.1, it can be seen that turbine work output drops from 76.1 kW to 74.9 kW per kg of basic solution and cooling capacity drops from 26.0 kW to 24.8 kW per kg of basic solution.

Finally, the overall effect of the irreversibilities associated with the cycle was analyzed. The result is shown in table 3.8. First law cycle efficiency drops from 23.27% under ideal condition to 18.52%. Turbine work output drops from 76.1 to 67.1 kW per kg of basic solution. Cooling capacity drops from 26.0 kW to 16.2 kW per kg of basic solution. The result shows that the irreversibilities associated with the cycle have considerable negative effect on the cycle performance.

Table 3.1 Typical Working Condition

Point	T(K)	P(bar)	H(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.1	30.0	-205.6	-0.0685	0.5300	1.0000
3	377.4	30.0	244.4	1.3072	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	257.0	2.0	1148.7	4.5556	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.1	30.0	-108.6	0.2655	0.4147	0.8003
12	299.3	2.0	-108.6	0.2763	0.4147	0.8003

Boiler heat input:	392.8 kJ/s
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	377.4 kJ/s
Turbine work output:	76.1 kW
Vapor quality at turbine exit:	93.10 %
Pump work input:	3.4 kW
Cooling capacity:	26.0 kW
Total heat input:	424.1 kJ/s
Total work output:	72.71 kW
First law cycle efficiency:	23.27 %

Table 3.2 Effect of Turbine Inefficiency

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.1	30.0	-205.6	-0.0685	0.5300	1.0000
3	377.4	30.0	244.4	1.3072	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	259.1	2.0	1186.8	4.7034	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.1	30.0	-108.6	0.2655	0.4147	0.8003
12	299.3	2.0	-108.6	0.2763	0.4147	0.8003

Boiler heat input:	392.8 kJ/s
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	377.4 kJ/s
Turbine work output:	68.5 kW
Vapor quality at turbine exit:	95.59 %
Turbine efficiency:	**90 %**
Pump work input:	3.4 kW
Cooling capacity:	18.4 kW
Total heat input:	424.1 kJ/s
Total work output:	65.10 kW
First law cycle efficiency:	19.68 %

Table 3.3 Effect of Pump Inefficiency

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.3	30.0	-204.7	-0.0655	0.5300	1.0000
3	377.4	30.0	244.6	1.3077	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	257.0	2.0	1148.7	4.5556	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.3	30.0	-107.8	0.2682	0.4147	0.8003
12	299.3	2.0	-107.8	0.2791	0.4147	0.8003

Boiler heat input:	392.6 kJ/s
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	378.0 kJ/s
Turbine work output:	76.1 kW
Vapor quality at turbine exit:	93.10 %
Pump work input:	4.2 kW
Pump efficiency:	**80 %**
Cooling capacity:	26.0 kW
Total heat input:	423.9 kJ/s
Total work output:	71.86 kW
First law cycle efficiency:	23.08 %

Table 3.4 Effect of 5% Pressure Loss in the Boiler

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.1	31.6	-205.4	-0.0685	0.5300	1.0000
3	377.4	31.6	244.6	1.3070	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	257.0	2.0	1148.7	4.5556	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.1	30.0	-108.6	0.2656	0.4147	0.8003
12	299.3	2.0	-108.6	0.2764	0.4147	0.8003

Boiler heat input:	392.7 kJ/s
Boiler pressure loss:	**5 %**
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	377.4 kJ/s
Turbine work output:	76.1 kW
Vapor quality at turbine exit:	93.10 %
Pump work input:	3.6 kW
Cooling capacity:	26.0 kW
Total heat input:	423.9 kJ/s
Total work output:	72.51 kW
First law cycle efficiency:	23.24 %

Table 3.5 Effect of 5% Pressure Loss in the Superheater

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.1	30.0	-205.6	-0.0685	0.5300	1.0000
3	377.4	30.0	244.4	1.3072	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	28.5	1533.8	4.5880	0.9921	0.1997
8	257.3	2.0	1157.0	4.5880	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.1	30.0	-108.6	0.2655	0.4147	0.8003
12	299.3	2.0	-108.6	0.2763	0.4147	0.8003

Boiler heat input:	392.8 kJ/s
Superheat input:	32.1 kJ/s
Superheater pressure loss:	**5 %**
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	377.4 kJ/s
Turbine work output:	75.2 kW
Vapor quality at turbine exit:	93.68 %
Pump work input:	3.4 kW
Cooling capacity:	24.3 kW
Total heat input:	424.9 kJ/s
Total work output:	71.86 kW
First law cycle efficiency:	22.64 %

Table 3.6 Effect of 5% Pressure Loss in the Solution Heat Exchanger

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.1	31.6	-205.4	-0.0685	0.5300	1.0000
3	377.5	30.0	244.7	1.3079	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	257.0	2.0	1148.7	4.5556	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.1	28.5	-108.7	0.2657	0.4147	0.8003
12	299.3	2.0	-108.7	0.2760	0.4147	0.8003

Boiler heat input:	392.5 kJ/s
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	377.3 kJ/s
Solution H.E. pressure loss:	**5 %**
Turbine work output:	76.1 kW
Vapor quality at turbine exit:	93.10 %
Pump work input:	3.6 kW
Cooling capacity:	26.0 kW
Total heat input:	423.8 kJ/s
Total work output:	72.51 kW
First law cycle efficiency:	23.24 %

Table 3.7 Effect of 5% Pressure Loss in the Cooler

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.1	30.0	-205.6	-0.0685	0.5300	1.0000
3	377.4	30.0	244.4	1.3072	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	30.0	1529.7	4.5556	0.9921	0.1997
8	258.3	2.1	1154.4	4.5556	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.1	30.0	-108.6	0.2655	0.4147	0.8003
12	299.3	2.0	-108.6	0.2763	0.4147	0.8003

Boiler heat input:	392.8 kJ/s
Superheat input:	31.3 kJ/s
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	377.4 kJ/s
Turbine work output:	74.9 kW
Vapor quality at turbine exit:	95.00 %
Pump work input:	3.4 kW
Cooling capacity:	24.8 kW
Cooler pressure loss:	**5 %**
Total heat input:	424.1 kJ/s
Total work output:	71.55 kW
First law cycle efficiency:	22.73 %

Table 3.8 Combined Effect of the Cycle Irreversibilities

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	280.0	2.0	-208.9	-0.0685	0.5300	1.0000
2	280.3	33.2	-204.2	-0.0651	0.5300	1.0000
3	377.5	31.6	245.1	1.3084	0.5300	1.0000
4	400.0	30.0	1547.2	4.6102	0.9432	0.2363
5	360.0	30.0	207.2	1.1414	0.6763	0.0366
6	360.0	30.0	1373.2	4.1519	0.9921	0.1997
7	410.0	28.5	1533.8	4.5880	0.9921	0.1997
8	260.8	2.1	1197.8	4.7320	0.9921	0.1997
9	280.0	2.0	1278.8	5.0467	0.9921	0.1997
10	400.0	30.0	349.0	1.5778	0.4147	0.8003
11	300.3	28.5	-107.8	0.2689	0.4147	0.8003
12	299.3	2.0	-107.8	0.2791	0.4147	0.8003

Boiler heat input:	392.2 kJ/s
Boiler pressure loss:	**5 %**
Superheat input:	32.1 kJ/s
Superheater pressure loss:	**5 %**
Rectifier heat rejection:	83.8 kJ/s
Absorber heat rejection:	378.0 kJ/s
Solution H.E. pressure loss:	**5 %**
Turbine work output:	67.1 kW
Vapor quality at turbine exit:	96.43 %
Turbine efficiency:	**90 %**
Pump work input:	4.7 kW
Pump efficiency:	**80 %**
Cooling capacity:	16.2 kW
Cooler pressure loss:	**3 %**
Total heat input:	424.2 kJ/s
Total work output:	62.38 kW
First law cycle efficiency:	18.52 %

CHAPTER 4
OPTIMIZATION OF AMMONIA-BASED COMBINED POWER/COOLING CYCLE

In general, optimization is a way to find the best solution to a problem of any kind. We do a lot of optimization in our daily life. For example, we always search for the best product, which meets our demands, with the lowest possible price. In engineering, optimization is a very powerful tool. There is usually more than one acceptable solution to an engineering problem. The goal of an engineer is to find the best one. When there are more than a few acceptable solutions, it may be simply impossible to compare them by hand. Sophisticated mathematical methods have been developed to help find the best solution.

Introduction to Optimization

Mathematical Formulation

Mathematically speaking, optimization is the minimization or maximization of a function subject to a set of constraints on its variables. For a thermodynamic cycle, the optimization objective usually is to find the maximum efficiency or work output. Engineering optimization problems, despite their diversity, have amazingly the same mathematical formulation. It can be written as:

$$\begin{aligned} &\min \quad f(\boldsymbol{x}) \\ &s.t. \quad \boldsymbol{h}(\boldsymbol{x}) = 0 \\ &\quad\;\; \boldsymbol{g}(\boldsymbol{x}) \le 0 \end{aligned} \tag{4.1}$$

Where

$\boldsymbol{x}$ is the vector of a set of continuous real *variables*;

f is the *objective function*, a function of $\boldsymbol{x}$ that we want to minimize;

$\boldsymbol{h}(\boldsymbol{x}) = 0$ and $\boldsymbol{g}(\boldsymbol{x}) \leq 0$ are sets of *equality constraints* and *inequality constraints*, respectively.

If an optimization problem is to maximize rather than minimize f, we can easily accommodate this change by minimizing $-f$ in the formulation (4.1). A well-defined optimization problem features a number of variables greater than the number of equality constraints, which implies that there exist degrees of freedom upon which we optimize.

Formulation (4.1) contains a number of classes of optimization problems, by appropriate consideration or elimination of its elements. If a problem does not have constraints on the variables, then formulation (4.1) becomes an *unconstrained optimization* problem. Otherwise it is a *constrained optimization* problem. Among constrained optimization problems, if both the objective function and all the constraints are linear functions of $\boldsymbol{x}$, then formulation (4.1) becomes a *linear programming* (LP) problem. If at least one of the constraints or the objective function is a nonlinear function, formulation (4.1) becomes a *nonlinear programming* (NLP) problem. A special case of constrained optimization is bound-constrained optimization. In this special case, all constraints are boundary constraints on $\boldsymbol{x}$ ($\boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U}$).

Optimality Conditions

Before we start to search for the optimum, we need to answer a very basic question. How do we tell if a point is an optimum or not? Apparently, a condition or conditions are required to make such judgement. That condition or conditions are termed

as optimality conditions. This section presents the necessary and sufficient optimality conditions for unconstrained and constrained optimization.

Unconstrained optimization

An unconstrained optimization problem deals with the search for a minimum of a function $f(\boldsymbol{x})$ of n real variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. For an unconstrained optimization problem, formulation (4.1) is simplified as:

$$\min \quad f(\boldsymbol{x}) \tag{4.2}$$

Each of the real variables x_i *is* allowed to take any value from $-\infty$ to ∞.

Unconstrained optimization plays an important role in the optimization applications. It is not only because there are some applications which inherently are unconstrained, but also because constrained optimization algorithms are often the extensions of unconstrained optimization algorithms. The necessary and sufficient optimality conditions for an unconstrained optimization problem are:

Theorem 4.1

Let $f(\boldsymbol{x})$ be a *twice differentiable function* in R^n at $\boldsymbol{x}^*$. Then $\boldsymbol{x}^*$ is a *local minimum* if and only if

i. $\nabla f(\boldsymbol{x}^*) = 0$, and

ii. The *Hessian matrix H(x*)*, given by

$$H(\boldsymbol{x}^*) = \frac{\partial^2 f}{\partial x_i \partial x_j}(\boldsymbol{x}^*) \tag{4.3}$$

is positive semidefinite; that is, $y^T H(\boldsymbol{x}^*) y \geq 0$ for all $y \in R^n$.

When there is only one variable, the above necessary and sufficient conditions are simplified into a more familiar format:

i. $f'(x) = 0$;

ii. $f''(x) \geq 0$.

Constrained optimization

A constrained optimization problem deals with the search for a minimum of a function $f(\boldsymbol{x})$ of n real variables $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ subject to a set of equality constraints $\boldsymbol{h}(\boldsymbol{x}) = 0$ ($h_i(\boldsymbol{x}) = 0$, $i = 1, 2, \cdots, m$), and a set of inequality constraints $\boldsymbol{g}(\boldsymbol{x}) \leq 0$ ($g_j(\boldsymbol{x}) \leq 0$, $j = 1, 2, \cdots, p$). It has the same form of formulation (4.1), which is rewritten here:

$$\begin{aligned} &\min \quad f(\boldsymbol{x}) \\ &s.t. \quad \boldsymbol{h}(\boldsymbol{x}) = 0 \\ &\qquad \boldsymbol{g}(\boldsymbol{x}) \leq 0 \end{aligned} \tag{4.1}$$

If a point $\bar{\boldsymbol{x}}$ satisfies the equality and inequality constraints in formulation (4.1), then it is a feasible point of the problem. Among inequality constraints, an inequality constraint $g_j(\boldsymbol{x})$ is called *active* at a feasible point $\bar{\boldsymbol{x}}$ if $g_j(\bar{\boldsymbol{x}}) = 0$ and *inactive* if $g_j(\bar{\boldsymbol{x}}) < 0$.

Compared with unconstrained optimization, constrained optimization is much more complicated. The key idea in developing necessary and sufficient optimality conditions for constrained optimization problems is to transform them into unconstrained problems and then apply the optimality conditions of unconstrained problems (Nocedal and Wright, 1999). The transformation involves the introduction of an auxiliary function, called the Lagrange function $L(\boldsymbol{x})$, defined as follows:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{g}(\boldsymbol{x}), \qquad \boldsymbol{\mu} \geq 0, \tag{4.4}$$

Where $\lambda^T = (\lambda_1, \lambda_2, \cdots, \lambda_m)$ and $\mu^T = (\mu_1, \mu_2, \cdots, \mu_p)$ are the Lagrange multipliers associated with the equality and inequality constraints, respectively. The multipliers λ can be any real numbers, while the multipliers μ must be nonnegative.

Lagrange multipliers in formulation (4.4) are not just a group of coefficients. They have an important special meaning in constrained optimization. Each Lagrange multiplier for a given constraint indicates how much the Lagrange function L will change for a differential change of that constraint. Considering a simple problem with only one equality constraint:

$$\min \quad f(\boldsymbol{x})$$

$$s.t. \quad h(\boldsymbol{x}) = e$$

Then its Lagrange function $L(\boldsymbol{x}, \lambda)$ is:

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) + \lambda[h(\boldsymbol{x}) - e]$$

Take a derivative of the Lagrange function $L(\boldsymbol{x}, \lambda)$ with respect to e:

$$\frac{\partial L}{\partial e} = -\lambda$$

Therefore, the change of L with the constant e is proportional to the Lagrange multiplier. Since the Lagrange function L is equal to the objective function f at the optimum, we can say that the Lagrange multipliers indicate the sensitivity of the objective function $f(\boldsymbol{x})$ with respect to perturbations on the constraints.

With the introduction of Lagrange function, a constrained optimization problem is transformed into an unconstrained problem according to duality theory (Floudas, 1995).

$$\min_{x,\lambda,\mu \geq 0} L(\boldsymbol{x}, \lambda, \mu) = f(\boldsymbol{x}) + \lambda^T \boldsymbol{h}(\boldsymbol{x}) + \mu^T \boldsymbol{g}(\boldsymbol{x}) \tag{4.5}$$

The necessary optimality conditions are given as below:

Theorem 4.2 Karush-Kuhn-Tucker Necessary Conditions

Let $f(x)$ and $g(x)$ be differentiable at x^* and $h(x)$ have continuous first partial derivatives at x^*. If x^* is a local optimum of problem (4.1) and one constraint qualification[1] is satisfied. Then there exist Lagrange multipliers λ, μ:

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) + \mu^T \nabla g(x^*) = 0$$

$$h(x^*) = 0$$

$$g(x^*) \le 0$$

$$\mu_j g_j(x^*) = 0 \qquad j = 1, 2, \cdots, p$$

$$\mu_j \ge 0 \qquad j = 1, 2, \cdots, p.$$

A point which satisfies Karush-Kuhn-Tucker conditions is called a KKT point.

However, theorem 4.2 only utilizes the first-order information of the objective function and constraints. The second order information, the curvature of the functions, is not considered. Fiacco and McCormick (1968) demonstrated that first-order information is not complete with their famous example:

$$\min \qquad (x_1 - 1)^2 + x_2^{\,2}$$

$$s.t. \qquad x_1 - \frac{x_2^2}{k} \le 0$$

Where the values of the parameter $k > 0$, for which (0,0) is a local minimum, are sought.

In this example, we only have one constraint. Its gradient

$$(1, -\frac{2x_2}{k})$$

[1] First-order and second-order constraint qualifications are satisfied if the gradients of all equality and active inequality constraints are linearly independent.

is always nonzero. So it is linearly independent by itself. A constrain qualification is satisfied. At (0,0), the gradient KKT conditions are:

$$\begin{pmatrix} -2 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\therefore \qquad \mu = 2 \ \geq 0$$

The constraints KKT conditions are satisfied since:

$$x_1 - \frac{x_2^2}{k} = 0$$

Therefore, KKT necessary conditions are satisfied. However, for $k = 1$ the point (0,0) is not a local minimum while for $k = 4$ it is.

From the above example, we learn that in order to find a local optimum, second order information has to be taken into account. The second order necessary and sufficient conditions are presented:

Theorem 4.3 Second-Order Necessary Optimality Conditions

Let $f(\boldsymbol{x})$, $\boldsymbol{h}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ be twice continuously differentiable at $\boldsymbol{x}^*$. If $\boldsymbol{x}^*$ is a local optimum of problem (4.1) and the second order constraint qualification is satisfied[1], and if there exist Lagrange multipliers $\boldsymbol{\lambda}^*$, $\boldsymbol{\mu}^*$ satisfying the KKT first-order necessary conditions:

$$\nabla f(\boldsymbol{x}^*) + \boldsymbol{\lambda}^{*T} \nabla \boldsymbol{h}(\boldsymbol{x}^*) + \boldsymbol{\mu}^{*T} \nabla \boldsymbol{g}(\boldsymbol{x}^*) = 0$$

$$\boldsymbol{h}(\boldsymbol{x}^*) = 0$$

$$\boldsymbol{g}(\boldsymbol{x}^*) \leq 0$$

$$\mu_j^* g_j(\boldsymbol{x}^*) = 0 \qquad j = 1, 2, \cdots, p$$

$$\mu_j^* \geq 0 \qquad j = 1, 2, \cdots, p$$

and if for every nonzero vector z:

$$\begin{cases} z^T \nabla h_i(x^*) = 0 & i = 1, 2, \cdots, m, \text{ and} \\ z^T \nabla g_i(x^*) = 0 & j \in J \equiv \{j : g_j(x^*) = 0\} \end{cases}$$

then ,

$$z^T \nabla^2 L(x^*, \lambda^*, \mu^*) z \geq 0$$

Theorem 4.4 Second-Order Sufficient Optimality Conditions

Let $f(x)$, $h(x)$ and $g(x)$ be twice differentiable at x^*. If there exist Lagrange multipliers λ^*, μ^* satisfying the KKT first-order necessary conditions:

$$\nabla f(x^*) + \lambda^{*T} \nabla h(x^*) + \mu^{*T} \nabla g(x^*) = 0$$

$$h(x^*) = 0$$

$$g(x^*) \leq 0$$

$$\mu_j^* g_j(x^*) = 0 \qquad j = 1, 2, \cdots, p$$

$$\mu_j^* \geq 0 \qquad j = 1, 2, \cdots, p$$

If for every nonzero vector z:

$$\begin{cases} z^T \nabla h_i(x^*) = 0 & i = 1, 2, \cdots, m, \text{ and} \\ z^T \nabla g_i(x^*) = 0 & j \in J_1 \equiv \{j : g_j(x^*) = 0, \ \mu_j > 0\}, \\ z^T \nabla g_i(x^*) \leq 0 & j \in J_2 \equiv \{j : g_j(x^*) = 0 \ \ \mu_j = 0\}, \end{cases}$$

it follows that

$$z^T \nabla^2 L(x^*, \lambda^*, \mu^*) z > 0,$$

then, x^* is a strict local minimum of problem (4.1).

Generalized Reduced Gradient Algorithm

The optimality conditions presented in the previous section provide the theoretical basis for the development of different optimization algorithms. Generalized Reduced Gradient (GRG) Algorithm is one of the most commonly used algorithms for solving constrained optimization.

As stated in the previous section, a constrained optimization problem can be expressed as:

$$\begin{aligned} &\min \quad f(\boldsymbol{x}) \\ &s.t. \quad \boldsymbol{h}(\boldsymbol{x}) = 0 \\ &\qquad \boldsymbol{g}(\boldsymbol{x}) \le 0 \end{aligned} \tag{4.1}$$

Among inequality constraints, they are subdivided into two categories: boundary constraints ($L_i \le x_i \le U_i$, one of L_i, U_i could be infinite) and function inequality constraints. Boundary constraints are separated from other inequality constraints because they are treated differently in determining the step length in a search direction, as we will see later. Function inequality constraints can be transformed into equality constraints by adding slack variables. For example, an inequality constraint

$$g_j(\boldsymbol{x}) \le 0$$

By adding slack variable x_{n+j}, it can be redefined as

$$h_{m+j}(\boldsymbol{x}) = g_j(\boldsymbol{x}) + x_{n+j} = 0$$

Where $x_{n+j} \ge 0$. Contrary to slack variables, variable x_1, x_2, …, x_n are called natural variables. Therefore, an inequality constraint, which is the function of natural variables, is transformed into an equality constraint, which is the function of natural variables plus a slack variable, and a boundary constraint on that slack variable.

By treating boundary constraints separately and transforming inequality constraints to equality constraints, problem (4.1) is converted to the following equivalent form:

$$\begin{array}{lll} \min & f(\boldsymbol{x}) & \boldsymbol{x} = [x_1, x_2, \cdots, x_n]^T \\ s.t. & h_j(\boldsymbol{x}) = 0 & j = 1, \cdots, m \\ & L_i \le x_i \le U_i & i = 1, \cdots, n \end{array} \tag{4.6}$$

However, keep in mind that m, n are different numbers here with those in formulation (4.1) and $\boldsymbol{x}$ is the set of natural variables and slack variables. (4.6) is the formulation we are going to use for describing GRG algorithm.

The generalized reduced gradient algorithm is based on a very basic optimization procedure which transforms a constrained optimization problem into one that is unconstrained or bound-constrained. It classes the variables as independent and dependent, and substitutes into the objective function the expressions obtained from the equality constraints in independent variables for the dependent variables. This essentially reduces the original problem to an unconstrained or bound-constrained problem of reduced dimension (Lasdon, et al., 1978).

In terms of dependent and independent variables, the constraints in formulation (4.6) can be written as

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}_I, \boldsymbol{x}_D) = 0$$

Where $\boldsymbol{x}_I$ are the set of independent variables, $\boldsymbol{x}_D$ are the set of dependent variables. $\boldsymbol{x}_D$ are solved such that

$$\boldsymbol{x}_D = \varphi(\boldsymbol{x}_I)$$

According to the formulation (4.6), the number of $\boldsymbol{x}_I$ is (n-m) and $\boldsymbol{x}_D$ is m. Thus the objective function can be written as

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_I, \boldsymbol{x}_D) = f(\boldsymbol{x}_I, \varphi(\boldsymbol{x}_I)) = F(\boldsymbol{x}_I)$$

The original problem is now transformed into a simpler reduced problem of (n-m) dimension:

$$\begin{aligned} &\min \quad F(\boldsymbol{x}) \\ &\text{s.t.} \quad \boldsymbol{L} \le \boldsymbol{x} \le \boldsymbol{U} \end{aligned} \tag{4.7}$$

Where $\boldsymbol{L}$ and $\boldsymbol{U}$ are the bound vectors for $\boldsymbol{x}$. The function $F(\boldsymbol{x})$ is called the reduced objective.

The concept of the reduced gradient is best described by the following simple example (Edgar and Himmelblau, 1988):

$$\begin{aligned} &\min \quad f(x_1, x_2) \\ &\text{s.t.} \quad h(x_1, x_2) = 0 \end{aligned} \tag{4.8}$$

The total derivatives of each function are:

$$df(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x_1} dx_1 + \frac{\partial f(\boldsymbol{x})}{\partial x_2} dx_2 \tag{4.9}$$

$$dh(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial x_1} dx_1 + \frac{\partial h(\boldsymbol{x})}{\partial x_2} dx_2 = 0 \tag{4.10}$$

If x_1 is designated to be the dependent variable, x_2 to be the independent variable, then dx_1 can be eliminated from Eq. (4.9) by use of Eq. (4.10) since

$$dx_1 = -\frac{[\partial h(\boldsymbol{x})/\partial x_2]}{[\partial h(\boldsymbol{x})/\partial x_1]} dx_2 \tag{4.11}$$

Therefore,

$$df(\boldsymbol{x}) = \{[\frac{\partial f(\boldsymbol{x})}{\partial x_2}] - [\frac{\partial f(\boldsymbol{x})}{\partial x_1}][\frac{\partial h(\boldsymbol{x})}{\partial x_1}]^{-1}[\frac{\partial h(\boldsymbol{x})}{\partial x_2}]\}dx_2 \tag{4.12}$$

The expression in the brace is called the *reduced gradient.* Of course, in this problem the reduced gradient contains only one element because there is only one independent variable. The general form of the reduced gradient is:

$$g_R^T = [\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_\mathbf{I}}]^T - [\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_\mathbf{D}}]^T[\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}_\mathbf{D}}]^{-1}[\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}_\mathbf{I}}] \tag{4.13}$$

The generalized reduced-gradient algorithm can be stated in five basic steps (Edgar and Himmelblau, 1988). In the text, $\boldsymbol{x}_\mathbf{I}$ and $\boldsymbol{x}_\mathbf{D}$ bear superscripts k since their components may well change from stage to stage. Also to simplify the notation, we denote $f(\boldsymbol{x}^k)$ as f^k and $\boldsymbol{h}(\boldsymbol{x}^k)$ as $\boldsymbol{h}^k$.

Step 1 Determine the search components for the independent variables.

At stage k, linearize the constraints at the feasible point $\boldsymbol{x}^k$ and compute the reduced gradient:

$$(g_R{}^k)^T = [\frac{\partial f^k}{\partial \boldsymbol{x}_\mathbf{I}^k}]^T - [\frac{\partial f^k}{\partial \boldsymbol{x}_\mathbf{D}^k}]^T[\frac{\partial \boldsymbol{h}^k}{\partial \boldsymbol{x}_\mathbf{D}^k}]^{-1}[\frac{\partial \boldsymbol{h}^k}{\partial \boldsymbol{x}_\mathbf{I}^k}] \tag{4.14}$$

Then the search direction components of the independent variables are established as:

$$\Delta_i^k = -g_{Ri}^k \tag{4.15}$$

However, Δ_i^k is selected as 0 if x_i is at one of its bounds and the step would exceed the bound. That is,

$$\Delta_i^k = 0 \qquad \text{if } x_i^k = U_i \quad g_{Ri} < 0 \qquad = L_i \quad g_{Ri} > 0 \tag{4.16}$$

Step 2 Determine the search components for the dependent variables.

A desired search direction would be along the surface of the constraints. This is accomplished by finding the tangent to $\boldsymbol{h}(\boldsymbol{x}) = 0$ at the point $(\boldsymbol{x}_{\mathrm{I}}^{k}, \boldsymbol{x}_{\mathrm{D}}^{k})$. That is,

$$d\boldsymbol{h} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\mathrm{I}}} d\boldsymbol{x}_{\mathrm{I}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\mathrm{D}}} d\boldsymbol{x}_{\mathrm{D}} = 0$$

$$d\boldsymbol{x}_{\mathrm{D}} = -[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\mathrm{D}}}]^{-1}[\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{\mathrm{I}}}]d\boldsymbol{x}_{\mathrm{I}}$$

Therefore, the search direction components of the dependent variables are:

$$\Delta_{\mathrm{D}}^{k} = -[\frac{\partial \boldsymbol{h}^{k}}{\partial \boldsymbol{x}_{\mathrm{D}}^{k}}]^{-1}[\frac{\partial \boldsymbol{h}^{k}}{\partial \boldsymbol{x}_{\mathrm{I}}^{k}}]\Delta_{\mathrm{I}}^{k} \tag{4.17}$$

Step 3 Improving the value of the objective function.

Minimize $f(\boldsymbol{x}_{\mathrm{I}}^{k} + \lambda\Delta_{\mathrm{I}}^{k}, \boldsymbol{x}_{\mathrm{D}}^{k} + \lambda\Delta_{\mathrm{D}}^{k})$ with respect to $\lambda\,(\lambda > 0)$ by a one-dimensional search. The new point is:

$$\boldsymbol{x}_{\mathrm{I}}^{k+1} = \boldsymbol{x}_{\mathrm{I}}^{k} + \lambda^{k}\Delta_{\mathrm{I}}^{k} \tag{4.18}$$

$$\tilde{\boldsymbol{x}}_{\mathrm{D}}^{k+1} = \boldsymbol{x}_{\mathrm{D}}^{k} + \lambda^{k}\Delta_{\mathrm{D}}^{k} \qquad \tilde{\boldsymbol{x}}_{\mathrm{D}} \text{ is a tentative point}$$

Step 4 Use Newton's method to regain feasibility of dependent variables.

In general, some of the components of $\tilde{\boldsymbol{x}}_{\mathrm{D}}^{k+1}$ are not feasible, hence $\boldsymbol{h}(\boldsymbol{x}_{I}^{k+1}, \tilde{\boldsymbol{x}}_{D}^{k+1}) \neq 0$. In order to return from a point away from constraints back to a point satisfying the constraints, $\tilde{\boldsymbol{x}}_{\mathrm{D}}^{k+1}$ is modified by Newton's method:

$$\boldsymbol{x}_{\mathrm{D}}^{k+1} = \tilde{\boldsymbol{x}}_{\mathrm{D}}^{k+1} - [\frac{\partial \boldsymbol{h}(\boldsymbol{x}_{\mathrm{I}}^{k+1}, \tilde{\boldsymbol{x}}_{\mathrm{D}}^{k+1})}{\partial \boldsymbol{x}_{\mathrm{D}}^{k}}]^{-1}\boldsymbol{h}(\boldsymbol{x}_{\mathrm{I}}^{k+1}, \tilde{\boldsymbol{x}}_{\mathrm{D}}^{k+1}) \tag{4.19}$$

Several iterations of Newton's method may be needed to find a feasible point. Figure 4.1 illustrates the move from $\boldsymbol{x}^k$ to $(\boldsymbol{x}_I^{k+1}, \widetilde{\boldsymbol{x}}_D^{k+1})$ and then to $(\boldsymbol{x}_I^{k+1}, \boldsymbol{x}_D^{k+1})$ by Newton's method.

Note: Newton's method finds the desired root $\boldsymbol{x}_D^{k+1}$ with $\widetilde{\boldsymbol{x}}_D^{k+1}$ as the starting point. By expanding $\boldsymbol{h}(\boldsymbol{x}_I^{k+1}, \boldsymbol{x}_D^{k+1})$ and equating it to zero, we get

$$\boldsymbol{h}(\boldsymbol{x}_I^{k+1}, \boldsymbol{x}_D^{k+1}) \approx \boldsymbol{h}(\boldsymbol{x}_I^{k+1}, \widetilde{\boldsymbol{x}}_D^{k+1}) + \frac{\partial \boldsymbol{h}(\boldsymbol{x}_I^{k+1}, \widetilde{\boldsymbol{x}}_D^{k+1})}{\partial \boldsymbol{x}_D^k}(\boldsymbol{x}_D^{k+1} - \widetilde{\boldsymbol{x}}_D^{k+1}) = 0$$

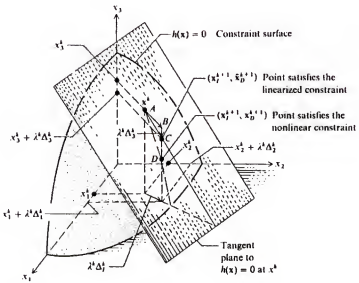


Figure 4.1 Illustration Of Using Newton's Method To Regain Feasibility

Step 5 Procedure on lack of convergence of Newton's method.

(a) If $\boldsymbol{x}^{k+1}$ is a feasible point and $f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k)$, adopt $\boldsymbol{x}^{k+1}$, and start with step 1 again.

(b) If $\boldsymbol{x}^{k+1}$ is a feasible point and $f(\boldsymbol{x}^{k+1}) > f(\boldsymbol{x}^k)$, reduce λ by a factor and start step 4 over again.

(c) If Newton's method fails to find a feasible point after certain times, reduce λ by a factor and start step 4 over again.

(d) If neither (a), (b) nor (c) occur, or (b) and (c) are unsuccessful after several passes, change the basis by exchanging a dependent variable with a former independent variable. This process is termed a *change of basis*.

Theoretically, the stopping condition for optimization is when KKT conditions are satisfied. However, in practice, we can use a simpler condition:

$$\left|\frac{f(\boldsymbol{x}^{k+1})-f(\boldsymbol{x}^{k})}{f(\boldsymbol{x}^{k})}\right|<\varepsilon \tag{4.20}$$

Where ε is a small number.

A numeric example is given as following to illustrate the GRG algorithm.

EXAMPLE:

$$\max \quad f(\boldsymbol{x})=(2x_1-0.5x_1^2)+(3x_2-0.5x_2^2)$$

$$s.t. \qquad h(\boldsymbol{x})=x_1^2+x_2^2+x_3-1=0$$

$$-2\le x_j\le 2 \qquad j=1,2,3$$

Solution: We minimize $-f(\boldsymbol{x})$. In the text below, $f(\boldsymbol{x})$ actually refers to $-f(\boldsymbol{x})$.

Step 1. Designate $\boldsymbol{x}_{\mathrm{I}}^{T}=[x_1 \quad x_3]$ and $\boldsymbol{x}_{\mathrm{D}}=[x_2]$. Pick [0.5, 0.5, 0.5] as the starting point. The reduced gradient is:

$$g_R^{T}=[\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{I}}}]^{T}-[\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{D}}}]^{T}[\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{D}}}]^{-1}[\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{I}}}]$$

where

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{I}}}=-\begin{bmatrix}2-x_1\\0\end{bmatrix} \qquad \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathrm{D}}}=-(3-x_2)$$

$$\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathbf{I}}} = [2x_1 \quad 1] \qquad \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}_{\mathbf{D}}} = 2x_2$$

$$(g_R^0)^T = -[2 - x_1^0 \quad 0] - [-(3 - x_2^0)\lambda]\left[\frac{1}{2x_2^0}\right][2x_1^0 \quad 1] = [1 \quad 2.5]$$

As the result, the search direction of the independent variable is,

$$\Delta_{\mathbf{I}}^0 = -g_R^0 = \begin{bmatrix} -1 \\ -2.5 \end{bmatrix}$$

Step 2. The search direction of the dependent variable is:

$$\Delta_{\mathbf{D}}^0 = -[\frac{\partial \boldsymbol{h}^0}{\partial \boldsymbol{x}_{\mathbf{D}}^0}]^{-1}[\frac{\partial \boldsymbol{h}^0}{\partial \boldsymbol{x}_{\mathbf{I}}^0}]\Delta_{\mathbf{I}}^0 = -\left[\frac{1}{2x_2^0}\right][2x_1^0 \quad 1]\Delta_{\mathbf{I}}^0 = 3.5$$

Step 3. From above, we get

$$\Delta^0 = \begin{bmatrix} -1 \\ 3.5 \\ -2.5 \end{bmatrix}$$

$$f(x^0 + \lambda\Delta^0) = -\{[2(0.5 - \lambda) - 0.5(0.5 - \lambda)^2] + [3(0.5 + 3.5\lambda) - 0.5(0.5 + 3.5\lambda)^2]\}$$

Minimize $f(x^0 + \lambda\Delta^0)$ with respect to λ by setting $(\partial f / \partial \lambda) = 0$ and solving the resulting equation. We get

$$\lambda^0 = 0.547$$

$$\boldsymbol{x}_{\mathbf{I}}^1 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} + 0.547\begin{bmatrix} -1 \\ -2.5 \end{bmatrix} = \begin{bmatrix} -0.047 \\ -0.867 \end{bmatrix}$$

$$\tilde{\boldsymbol{x}}_{\mathbf{D}}^1 = [0.5] + 0.547[3.5] = 2.415$$

Step 4. Check if $h(x) = 0$,

$$h^1 = (-0.047)^2 + (2.415)^2 + (-0.867) - 1 = 3.967 \neq 0$$

Constraint is not satisfied. Applying Newton's method to modify $\tilde{x}_2^1$:

$$\frac{\partial h^1}{\partial x_2^1} = 2\tilde{x}_2^1 = 4.829$$

$$\hat{x}_2^1 = 2.415 - \left(\frac{1}{4.829}\right)(3.967) = 1.594$$

Successively applying Newton's method, starting each time with the previously found $\hat{x}_2^1$, until a feasible x_2^1 is found. We give the result here:

x_2^1=1.366

Step 5.

$f(\boldsymbol{x}^0) = -2.25$

$f(\boldsymbol{x}^1) = -3.07$

$\therefore \quad f(\boldsymbol{x}^1) < f(\boldsymbol{x}^0)$

x^1 is a better point and adopted as new starting point. New search starts with step 1 again.

Normally, the starting point, x^0, is a feasible solution. But that is not necessary. If the starting point, x^0, is not feasible, the optimization is done in two phases. In phases I, x^0 is made feasible by adding artificial variables to satisfy the equality constraints, then forcing them to zero values thus finding a new feasible point (Hwang, et al., 1972). Let an artificial variable, x_{n+i}, satisfy the condition,

$$0 \le x_{n+i} \le \infty$$

If, at the starting point $\boldsymbol{x}^0$, the ith constraint violates the equality condition by being greater than zero, i.e., $h_i(\boldsymbol{x}^0) > 0$, it is made feasible by subtracting the artificial variable, x_{n+i},

$$h_i(\boldsymbol{x}^0) - x_{n+i} = 0$$

Likely, if $h_i(\boldsymbol{x}^0) < 0$, it is made feasible by adding the artificial variable,

$h_i(x^0)+ x_{n+i} = 0$

The objective function to be minimized in phase I becomes the sum of all artificial variables,

$$\min \sum_{i \in NE} x_{n+i} \qquad NE = \{i \mid h_i(x) \neq 0\}$$

The artificial variables are forced to zero by minimization. When all of the artificial variables are zero, the solution, x, becomes feasible. The second phase optimization begins with this new feasible solution.

Description of the Problem

Variable Temperature Heat Source

For a given heat source fluid and a fixed ambient temperature T_0, the maximum useful work (per unit weight of heat source fluid) is a function of only T_{hs}^{in}, the heat source temperature:

$$w_{max} = (h_{hs}^{in} - h_0) - T_0(s_{hs}^{in} - s_0) \tag{4.21}$$

Where h_{hs}^{in} is the inlet enthalpy of the heat source fluid;

s_{hs}^{in} is the inlet entropy of the heat source fluid;

h_0 is the enthalpy of the heat source fluid at ambient temperature;

s_0 is the entropy of the heat source fluid at ambient temperature.

A plot of this value is given in Figure 4.2. The heat source fluid in Figure 4.2 is saturated water when its temperature is above 100°C and subcooled water at 1 atm when the temperature is below 100°C.

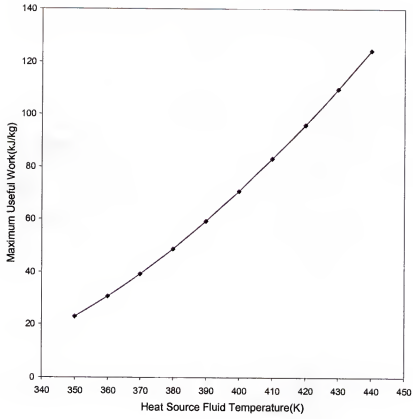


Figure 4.2 Maximum Useful Work Plotted as a Function of Heat Source Fluid Temperature

Any real process will have nonreversible steps that will result in a net work output less than maximum useful work. These irreversibilities can be evaluated for each step of the conversion process. For example, any heat exchange step destroys availability due to the finite temperature difference between the fluids. Although big temperature difference favors the heat transfer process, thus requiring less heat transfer surface area, it will cause a degradation of availability and a corresponding loss in thermodynamic performance of the process. Optimal operation with the minimum irreversibility occurs when the heat capacities of both streams are constant. This situation produces a balanced exchanger in which the temperature difference ΔT between streams can be kept uniform. In the heat exchanger, a "pinch" or minimum temperature difference occurs at the point where the

liquid begins to evaporate. Efficient resource utilization will result when (Milora and Tester, 1976):

1. Most of the heat is extracted from the heat source fluid before disposal.
2. Temperature differentials across heat transfer surfaces are maintained at minimum practical levels.
3. Turbines and feed pumps are carefully designed for optimum efficiency.
4. Heat is rejected from the thermodynamic cycle at a temperature near the minimum ambient temperature.

Quantitatively, one can define a cycle efficiency factor that is a measure of the fraction of the available energy of the heat source fluid that is converted into useful work. Second law thermal efficiency η_2, is defined in terms of net power output W_{net}, the heat source flow rate, m_{hs}, and the availability:

$$\eta_2 = \frac{W_{net}}{m_{hs}[(h_{hs}^{in} - h_0) - T_0(s_{hs}^{in} - s_0)]} \tag{4.22}$$

η_2 is a direct measure of the efficiency of resource utilization, because for a fixed T_{hs}^{in}, higher value of η_2 corresponds to lower flow rate of the source fluid for a given net power output. The second law efficiency, η_2, should not be confused with the first law thermal efficiency, η_1, which is a measure of how efficiently the transferred heat is converted into work regardless of the availability.

$$\eta_1 = \frac{W_{net}}{m_{hs}(h_{hs}^{in} - h_{hs}^{out})} \tag{4.23}$$

Where h_{hs}^{out} is the outlet enthalpy of the heat source fluid. The quality of the heat source and the availability of the disposed heat source fluid at temperatures above ambient are not considered in calculating η_1.

Optimization Model for the Cycle

Optimization is a very powerful tool in the engineering decision-making. A typical engineering problem can be expressed as follows: We have a process that can be represented by some equations and/or experimental data. We also have a performance criterion in mind, for example, to achieve maximum thermal efficiency. The goal of the optimization is to find an operating condition, which is represented by the values of the some variables, that yields the best performance.

Problem formulation is perhaps the most crucial step in an optimization problem. It requires identifying a conceptual statement of a given application, and organizing it into a standard mathematical form. Namely

1. The objective function
2. Constraints

The objective function represents, in terms of the key variables, the process being analyzed. The constraints describe the interrelationships of the key variables. Constraints in optimization problems arise from physical bounds on the variables, empirical relations, physical laws, etc.

For this ammonia-based combined power/cooling cycle, our optimization objective is to find the maximum thermal efficiency. By looking into the cycle (Fig. 2.7), we can see that the high and low pressure of the cycle can be freely changed and have to be chosen by the designer. In addition, the temperatures of the absorber, boiler, rectifier and superheater, are also free variables.

From the previous parametric analysis, we have concluded that higher concentration of the ammonia strong solution always results in a better cycle

performance. Therefore, a saturation state is desired in the absorber. For saturated ammonia solution, when its temperature and pressure are fixed, its concentration is also fixed. No additional variable is added into the set of the free variables.

In the boiler, the leaving ammonia weak solution is bound to be saturated and the leaving ammonia vapor is in an equilibrium state with the weak solution. So when the pressure and temperature of the boiler, and the concentration of the feeding ammonia strong solution are fixed, the concentration of the weak solution and ammonia vapor along with their fractions are determined.

For heat source, there are three unknown variables: m_{hs}, T_{hs}^{in}, T_{hs}^{out}. However, due to the energy balance in the boiler and superheater, the energy transferred out of the heat source equals the energy transferred into the cycle. So only two variables are actually unknown. Any two of them can be chosen as free variables. We choose T_{hs}^{in}, T_{hs}^{out}.

In summary, this ammonia-based combined power/cooling cycle has eight free variables: the absorber temperature T_{absorber}, the boiler temperature T_{boiler}, the rectifier temperature $T_{\text{rectifier}}$, the superheater temperature $T_{\text{superheater}}$, the high pressure P_{high}, the low pressure P_{low}, the heat source entrance temperature T_{hs}^{in}, and the heat source exit temperature T_{hs}^{out}. The cycle performance depends on the selection of the values for these eight free variables. Each combination of the eight values represents a distinctive operating condition of the cycle. Searching for optimum values for these variables is the task of this optimization work. Consequently, the objective function of the optimization of the cycle can be written as:

$$\eta = f(T_{\text{absorber}}, T_{\text{boiler}}, T_{\text{rectifier}}, T_{\text{superheater}}, P_{\text{high}}, P_{\text{low}}, T_{hs}^{in}, T_{hs}^{out}) \tag{4.24}$$

As we have discussed above, the second law thermal efficiency η_2 is the true reflection of the efficiency of resource utilization, therefore it is chosen as our optimization objective. For the ammonia-based combined power/cooling cycle, the second law thermal efficiency can be expressed as:

$$\eta_2 = \frac{W_t - W_p + Q_{cool}}{m_{hs}[(h_{hs}^{in} - h_0) - T_0(s_{hs}^{in} - s_0)]} \tag{4.25}$$

After we have decided on the objective function, the next step is to find the constraints, which are required for the practicability of the cycle. These constraints define feasible region. By common sense, the temperature of the rectifier has to be lower than the temperatures of the boiler and superheater. Besides, the temperature of the boiler has to be greater than a minimum value, which is the bubble point of the feeding ammonia solution. Otherwise, no ammonia will be boiled off from the ammonia solution. Similarly, the temperature of the rectifier has to be greater than a minimum value so ammonia vapor entering the rectifier unit will not be totally condensed.

For the turbine, lowering its exhaust pressure will increase the thermal efficiency. However, the moisture content of the fluid leaving the turbine is increased at the same time. This increased moisture content decreases the efficiency of an actual turbine. In addition, the impingement of liquid droplets on the turbine blades leads to a serious erosion problem. In practice, it is desirable to keep the moisture content less than about 10 percent at the low pressure end of the turbine (Wark, 1988). Therefore, the quality of the ammonia vapor at the exit of the turbine has to be greater than 90%.

In the cycle (shown in Figure 2.7), after being pumped to the high pressure, ammonia strong solution is separated into two streams (point 2). By common sense, their fractions are between 0 and 1. Of course, when either of them is between 0 and 1, the

other must be. So only one constraint is added into the set of the constraints. The fraction of the solution at point 2 separated into the stream at point 2'', $f_{2''}$, is randomly picked. In addition, the flow rate of the ammonia vapor leaving the boiler has to be less than that of feeding ammonia solution. Therefore, the fraction flowing through point 4, f_4, which is defined by m_4/m_1, is between 0 and 1.

Finally, the most common type of constraints in the thermodynamic cycle is the temperature difference limit on the heat exchangers. They are expressed mathematically with other constraints discussed as follows:

- $T_{\text{boiler}} \geq T_{\text{rectifier}}$
- $T_{\text{superheater}} \geq T_{\text{rectifier}}$
- $T_{\text{boiler}} \geq T_{\text{boilermin}}$
- $T_{\text{rectifier}} \geq T_{\text{rectifiermin}}$
- $x_{\text{turbine}} \geq 0.9$
- $0 \leq f_4 \leq 1$
- $0 \leq f_{2''} \leq 1$
- $T_{10} - T_{2'} \geq \Delta T_{min}$
- $T_{\text{I}} - T_{\text{superheater}} \geq \Delta T_{min}$
- $T_{\text{III}} - T_3 \geq \Delta T_{min}$
- $T_{\text{II}} - T_{\text{boiler}} \geq \Delta T_{min}$
- $\Delta T_{\text{pin}} \geq \Delta T_{pin}^{\min}$

Where

x_{turbine} is the quality of the vapor at the turbine exit;

T_x: temperature at point x (refer to Fig. 2.7);

ΔT_{min}: minimum temperature difference required in the heat exchangers;

ΔT_{pin}: temperature difference at the pinch point in the boiler;

$\Delta T_{pin}^{\min}$: minimum temperature difference required at the pinch point.

The variables that appear in the constraints are free variables or functions of the free variables. With an objective function and a complete set of constraints, the formulation of the optimization model for the ammonia-base combined power/cooling cycle is complete.

Optimization Program

For optimization, we use the GRG2 program written by Dr. Lasdon, which is by far the most widely distributed nonlinear programming code (More and Wright, 1993). It uses an implementation of the generalized reduced gradient (GRG) algorithm. It has both Fortran and C version. C version is used in the current research. In the program, the user is required to provide a subroutine *gcomp*. The values of the objective function and constraints are calculated in this subroutine and provided to the main optimization program. The user also provides data specifying the upper and lower bounds on the free variables and initial values of the free variables. In the subroutine *gcomp,* the thermodynamic calculation of the ammonia-based combined power/cooling cycle is done to get the objective function and constraints. Property programs are called to calculate the properties of ammonia/water mixture at various state points.

Optimization Results

In this section, optimization results are presented. Two typical heat source temperatures are studied. One is 440K, and the other is 360K. The ambient temperature for both cases is assumed as 290K. Their optimum working conditions are presented.

Resource Temperature of 440K

This heat source temperature is within the range of geothermal source and a CPC solar collector. The cycle is optimized for second law thermal efficiency. The optimum working conditions are presented in tables 4.1 and 4.2. The state points in the first column of the table 4.1 correspond to the locations given in figure 2.7.

Table 4.1 Optimum Working Conditions for a Source Temperature of 440K

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	8.6	66.0	0.3712	0.9500	1.0000
2	296.1	52.0	72.9	0.3712	0.9500	1.0000
3	313.4	52.0	156.2	0.6447	0.9500	1.0000
4	374.3	52.0	1315.6	3.8000	0.9943	0.7720
5	374.3	52.0	358.2	1.4403	0.7999	0.0000
6	374.3	52.0	1315.6	3.8000	0.9943	0.7720
7	435.0	52.0	1547.1	4.3606	0.9943	0.7720
8	300.5	8.6	1270.9	4.3606	0.9943	0.7720
9	300.5	8.6	1270.9	4.3606	0.9943	0.7720
10	374.3	52.0	358.2	1.4403	0.7999	0.2280
11	301.1	52.0	-7.3	0.3568	0.7999	0.2280
12	301.2	8.6	-7.3	0.3775	0.7999	0.2280

Table 4.2 Cycle Performance Parameters For Conditions In Table 4.1

Boiler Heat Input	941.1 kJ/s
Superheat Input	178.7 kJ/s
Absorber Heat Rejection	913.5 kJ/s
Turbine Work Output	213.3 kW
Vapor Quality at Turbine Exit	97.55 %
Pump Work Input	6.9 kW
Total Heat Input	1119.8 kJ/s
Total Work Output	206.37 kW
First Law Efficiency	18.43 %
Heat Source Flow Rate	2.783 kg/s
Heat Source Entrance Temperature	440 K
Heat Source Exit Temperature	345.4 K
Second law efficiency	59.63 %

It is noticed that the optimum concentration for ammonia strong solution is at 0.95. Though unnecessary, since the optimization program searches the entire

concentration range to find the optimum concentration, another optimization is done at a fixed 0.9 concentration for ammonia strong solution to verify the previous optimization result. The result is given in tables 4.3 and 4.4, which show a lower second law efficiency.

Table 4.3 Optimum Working Conditions at X = 0.9

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	8.2	29.6	0.3438	0.9000	1.0000
2	295.8	47.1	35.5	0.3438	0.9000	1.0000
3	327.2	47.1	186.7	0.8293	0.9000	1.0000
4	371.4	47.1	1329.4	3.8689	0.9939	0.5669
5	371.4	47.1	326.4	1.3787	0.7771	0.0000
6	371.4	47.1	1329.4	3.8689	0.9939	0.5669
7	435.0	47.1	1559.5	4.4308	0.9939	0.5669
8	302.3	8.2	1286.1	4.4308	0.9939	0.5669
9	302.3	8.2	1286.1	4.4308	0.9939	0.5669
10	371.4	47.1	326.4	1.3787	0.7771	0.4331
11	300.8	47.1	-22.6	0.3393	0.7771	0.4331
12	301.0	8.2	-22.6	0.3575	0.7771	0.4331

Table 4.4 Cycle Performance Parameters For Conditions In Table 4.3

Boiler Heat Input	708.3 kJ/s
Superheat Input	130.4 kJ/s
Absorber Heat Rejection	689.7 kJ/s
Turbine Work Output	155.0 kW
Vapor Quality at Turbine Exit	98.16 %
Pump Work Input	5.9 kW
Total Heat Input	838.7 kJ/s
Total Work Output	149.07 kW
First Law Efficiency	17.77 %
Heat Source Flow Rate	2.072 kg/s
Heat Source Entrance Temperature	440 K
Heat Source Exit Temperature	344.8 K
Second law efficiency	57.86 %

At this heat source temperature, the optimum working conditions do not produce any refrigeration capacity. However, refrigeration capacity can be obtained at a non-optimum condition if necessary. Tables 4.5 and 4.6 present one of such conditions:

Table 4.5 A Non-Optimum Working Condition with Refrigeration Capacity

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	4.6	-125.9	0.1777	0.6131	1.0000
2	295.2	31.8	-122.4	0.1777	0.6131	1.0000
3	374.8	31.8	334.2	1.5179	0.6131	1.0000
4	403.0	31.8	1553.2	4.5993	0.9401	0.4078
5	363.5	31.8	222.0	1.1859	0.6699	0.0648
6	363.5	31.8	1377.5	4.1409	0.9911	0.3430
7	366.4	31.8	1387.7	4.1684	0.9911	0.3430
8	277.4	4.6	1139.2	4.1684	0.9911	0.3430
9	285.0	4.6	1246.6	4.5514	0.9911	0.3430
10	403.0	31.8	363.9	1.6145	0.4157	0.6570
11	300.2	31.8	-108.1	0.2668	0.4157	0.6570
12	300.8	4.6	-108.1	0.2774	0.4157	0.6570

Table 4.6 Cycle Performance Parameters For Conditions In Table 4.5

Boiler Heat Input	523.9 kJ/s
Superheat Input	3.5 kJ/s
Absorber Heat Rejection	482.5 kJ/s
Turbine Work Output	81.77 kW
Vapor Quality at Turbine Exit	90 %
Pump Work Input	3.5 kW
Refrigeration Capacity	36.8 kW
Total Heat Input	527.4 kJ/s
Total Work Output	81.77 kW
First Law Efficiency	22.49 %
Heat Source Flow Rate	2.211 kg/s
Heat Source Entrance Temperature	440 K
Heat Source Exit Temperature	384.4 K
Second law efficiency	43.13 %

Resource Temperature of 360K

This heat source temperature is within the range of flat-plate solar collectors and solar ponds. The cycle is optimized for second law thermal efficiency. The optimum working condition for this heat source is listed in tables 4.7 and 4.8. At this heat source temperature, the optimum working condition does carry refrigeration capacity and the optimum concentration for ammonia strong solution is at a medium value, 0.67.

Table 4.7 Optimum Working Conditions for a Source Temperature of 360K

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	5.5	-105.1	0.1989	0.6733	1.0000
2	295.1	13.0	-104.1	0.1989	0.6733	1.0000
3	325.2	13.0	45.7	0.6805	0.6733	1.0000
4	333.1	13.0	1371.9	4.4956	0.9935	0.2066
5	311.1	13.0	102.9	0.5923	0.8949	0.0110
6	311.1	13.0	1302.7	4.2793	0.9990	0.1956
7	311.1	13.0	1302.7	4.2793	0.9990	0.1956
8	280.6	5.5	1195.5	4.2793	0.9990	0.1956
9	285.0	5.5	1278.3	4.5730	0.9990	0.1956
10	333.1	13.0	45.2	0.7305	0.5942	0.8044
11	300.1	13.0	-106.8	0.2500	0.5942	0.8044
12	300.2	5.5	-106.8	0.2532	0.5942	0.8044

Table 4.8 Cycle Performance Parameters For Conditions In Table 4.7

Boiler Heat Input	272.9 kJ/s
Absorber Heat Rejection	269.1 kJ/s
Turbine Work Output	21.0 kW
Vapor Quality at Turbine Exit	93.93 %
Pump Work Input	1.0 kW
Refrigeration Capacity	16.2 kW
Total Heat Input	272.9 kJ/s
Total Work Output	19.99 kW
First Law Efficiency	13.26 %
Heat Source Flow Rate	2.183 kg/s
Heat Source Entrance Temperature	360 K
Heat Source Exit Temperature	330.2 K
Work Output Per Unit Mass of Heat Source Fluid	9.16 kW
Refrigeration Output Per Unit Mass of Heat Source Fluid	7.42 kW
Second law efficiency	54.22 %

Optimization With Different Objective Functions

Although second law efficiency is the natural optimization objective for the cycle when variable temperature heat sources are used, the optimization technique presented above can be used to optimize for any other performance parameter in the cycle, such as first law efficiency, work output, refrigeration output, etc. Some examples for 360 K heat source temperature are presented below.

Tables 4.9 and 4.10 give the optimum working conditions and cycle performance parameters based on maximum work output per unit mass of heat source fluid. Tables 4.11 and 4.12 give the same information based on maximum refrigeration output per unit mass of heat source fluid.

A comparison of the three optimization results, based on maximum second law efficiency, maximum work output and maximum refrigeration output, shows that the second law efficiency is 43.13% for the maximum work output and 53.56% for the maximum refrigeration output as compared to a maximum obtainable value of 54.22% (Table 4.8). Maximum obtainable work output per kilogram of heat source fluid is 13.19 kW as compared to a work output of 9.16 kW for maximum resource utilization (max second law efficiency). However, it is seen that optimization for maximum work output gives no refrigeration while a refrigeration output of 7.42 kW per kilogram of heat source fluid is obtained for max second law efficiency. It is seen from tables 4.11 and 4.12 that maximization for refrigeration capacity gives us results close to those for maximum second law efficiency.

Table 4.9 Optimum Working Conditions Based On Maximum Work Output

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	8.6	66.0	0.3712	0.9500	1.0000
2	295.3	19.6	67.7	0.3712	0.9500	1.0000
3	300.0	19.6	90.1	0.4463	0.9500	1.0000
4	355.0	19.6	1408.8	4.4265	0.9880	0.9117
5	355.0	19.6	140.4	1.0195	0.5580	0.0000
6	355.0	19.6	1408.8	4.4265	0.9880	0.9117
7	355.0	19.6	1408.8	4.4265	0.9880	0.9117
8	309.5	8.6	1289.1	4.4265	0.9880	0.9117
9	309.5	8.6	1289.1	4.4265	0.9880	0.9117
10	355.0	19.6	140.4	1.0195	0.5580	0.0883
11	300.3	19.6	-112.4	0.2467	0.5580	0.0883
12	300.5	8.6	-112.4	0.2513	0.5580	0.0883

Table 4.10 Cycle Performance Parameters For Conditions In Table 4.9

Boiler Heat Input	1206.7 kJ/s
Superheat Input	0 kJ/s
Absorber Heat Rejection	1099.3 kJ/s
Turbine Work Output	109.1 kW
Vapor Quality at Turbine Exit	97.10 %
Pump Work Input	1.8 kW
Refrigeration Capacity	0 kW
Total Heat Input	1206.7 kJ/s
Total Work Output	107.39 kW
First Law Efficiency	8.90 %
Heat Source Flow Rate:	8.144 kg/s
Heat Source Entrance Temperature:	360 K
Heat Source Exit Temperature:	324.6 K
Work Output Per Unit Mass Flow Rate of Heat Source Fluid	13.19 kW
Refrigeration Output Per Unit Mass Flow Rate of Heat Source Fluid	0 kW
Second law efficiency:	43.13 %

Table 4.11 Optimum Working Conditions Based On Maximum Refrigeration Output

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	5.1	-114.8	0.1885	0.6478	1.0000
2	295.1	12.9	-113.8	0.1885	0.6478	1.0000
3	327.2	12.9	43.4	0.6928	0.6478	1.0000
4	334.8	12.9	1377.9	4.5164	0.9927	0.1744
5	311.4	12.9	95.6	0.5881	0.8828	0.0093
6	311.4	12.9	1304.2	4.2869	0.9989	0.1651
7	311.4	12.9	1304.2	4.2869	0.9989	0.1651
8	278.5	5.1	1189.4	4.2869	0.9989	0.1651
9	285.0	5.1	1282.7	4.6198	0.9989	0.1651
10	334.8	12.9	49.2	0.7500	0.5784	0.8349
11	300.1	12.9	-110.3	0.2471	0.5784	0.8349
12	300.2	5.1	-110.3	0.2504	0.5784	0.8349

Table 4.12 Cycle Performance Parameters For Conditions In Table 4.11

Boiler Heat Input:	237.1 kJ/s
Superheat Input:	0 kJ/s
Absorber Heat Rejection:	234.5 kJ/s
Turbine Work Output:	19.0 kW
Vapor Quality at Turbine Exit:	93.65 %
Pump Work Input:	1.0 kW
Refrigeration Capacity:	15.4 kW
Total Heat Input:	237.1 kJ/s
Total Work Output:	17.95 kW
First Law Efficiency:	14.07 %
Heat Source Flow Rate:	2.038 kg/s
Heat Source Entrance Temperature:	360 K
Heat Source Exit Temperature:	332.2 K
Work Output Per Unit Mass Flow Rate of Heat Source Fluid	8.81 kW
Refrigeration Output Per Unit Mass Flow Rate of Heat Source Fluid	7.56 kW
Second law efficiency:	53.56 %

Effect of Ambient Temperature

In the above optimization, ambient temperature is set at 290K. A change of ambient temperature will certainly affect the cycle performance. In order to analyze the effect of the ambient temperature, the cycle has been optimized based on maximum second law efficiency at different ambient temperatures. The results are shown in figures 4.3 – 4.6. Figures 4.3 and 4.4 show the results for a source at a temperature of 360K, which is easily available using high efficiency flat plate collectors, some geothermal resources, or industrial waste heat. It is assumed that the heat source is water at an initial temperature of 360K. Figures 4.5 and 4.6 show the optimization results for a heat source as water at an initial temperature of 440K. This temperature is in the range of some geothermal sources and can be achieved by a solar energy collection system using CPC collectors or other low concentration solar collectors.

Figure 4.3 shows the variation in energy inputs and outputs with the ambient temperature under optimal conditions. The energy inputs and outputs in this figure have been converted to values per kilogram of heat source (water) at an initial temperature of 360K. As expected, the heat input from the source to the working fluid goes down as the ambient temperature goes up. The energy output goes down similarly. The availability or exergy of the source fluid also goes down with the increase in the ambient temperature but not as much as the energy output. Therefore, the second law efficiency goes down from 58.02% to 47.15% as the ambient temperature goes up from 280 to 310K (Figure 4.4). The first law efficiency goes down from 15.71% to 10.46% for the same ambient temperature range. It is seen from figure 4.3 that under optimum operating conditions for a heat source at 360K, the refrigeration output is almost equal to the power output.

However, for a higher heat source temperature of 440K, there is no refrigeration output under optimized conditions. All of the energy output comes out as power output, as shown in figure 4.5. It must be pointed out here that it is possible to obtain both refrigeration and power output even at this source temperature, but that would be at non-optimum operating conditions. For the source temperature of 440K, the second law efficiency varies very little (from 59.99% to 59.36%) as the ambient temperature goes up from 280 to 310K. It must be noted, however, that even though the second law efficiency remains constant in this case, the work output decreases because the exergy of the source fluid goes down as the ambient temperature goes up.

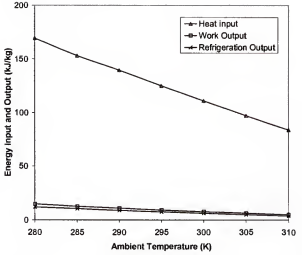


Figure 4.3 Variation of Energy Input and Output with Ambient Temperature (Per kg of water at 360K as Heat Source)

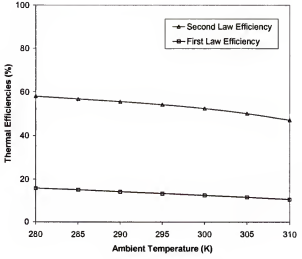


Figure 4.4 Variation of Thermal Efficiencies with Ambient Temperature (360K Heat Source Temperature)

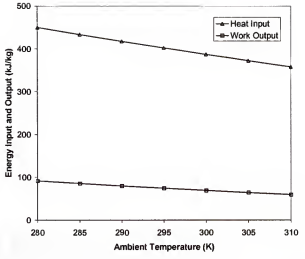


Figure 4.5 Variation of Energy Input and Output with Ambient Temperature
(Per kg of water at 440K as Heat Source)

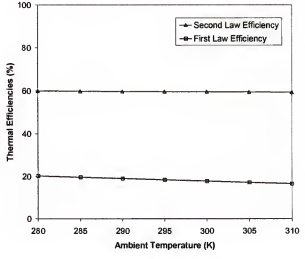


Figure 4.6 Variation of Thermal Efficiencies with Ambient Temperature
(440K Heat Source Temperature)

CHAPTER 5
APPLICATIONS OF THE NOVEL CYCLE

The ammonia-based combined power/cooling cycle uses ammonia-water mixture as the working fluid to obtain better thermal match between heat source and working fluid. It can produce power output as well as refrigeration output. In this chapter, some applications of the novel cycle are studied.

Solar Thermal Energy

The ammonia-based combined power/cooling cycle is a novel thermodynamic cycle, which can effectively utilize low temperature sensible heat sources, such as solar thermal energy, geothermal energy and waste heat. It uses ammonia-water mixture as working fluid to gain better thermal match between sensible heat source and working fluid. In the cycle, ammonia vapor generated in the boiler is purified in the rectifier and thus be able to expand to a low temperature in the turbine. Power as well as refrigeration could be produced from the cycle.

In this section, the application of the novel cycle for solar thermal energy conversion is studied. Solar energy is immense and renewable and considered as a future energy by many. There are two basic ways to convert solar energy to electricity: solar thermal conversion and photovoltaic conversion. By thermal conversion, solar radiation is converted to heat and then to mechanical energy by a thermodynamic cycle and finally to electricity through a generator. To collect solar radiation and convert it to heat, solar thermal collectors are used. There are a wide variety of solar collectors available today,

ranging from unglazed flat plate type solar collectors operating at about 5 - 10°C above the ambient to central receiver concentrating collectors operating at above 1000°C. Table 5.1 lists various types of solar thermal collectors and their typical temperature and concentration ranges (Goswami et al., 2000). While high concentration solar collectors give high working temperature, flat plate type or low concentration collectors have the advantage of low cost. The ammonia-based combined power/cooling cycle is able to use low cost solar, low concentration collectors and still gives satisfactory energy utilization effect.

Table 5.1 Types of Solar Thermal Collectors and Their Typical Temperature Range

Type of Collector	Concentration Ratio	Typical Working Temperature Range (°C)
Flat Plate Collector	1	≤70
High Efficiency Flat Plate Collector	1	60-120
Fixed Concentrator	3-5	100-150
Parabolic Trough Collector	10-50	150-350
Parabolic Dish Collector	200-500	250-700
Central Receiver	500->3000	500->1000

Figure 5.1 gives a schematic of a solar driven ammonia-based combined power/cooling system. It consists of two subsystems: a novel power/cooling cycle and a solar collector cycle that provides heat source for the novel power/cooling cycle. The heat source fluid (water) leaving the novel cycle is not discarded. Instead, it returns to the storage tank and is then re-heated before circulating through the novel cycle again. Flat plate or low concentration solar collectors may be used for this cycle. A collector heat exchanger and antifreeze are used in the system to prevent freezing in cold weather. To capitalize on whatever stratification may exist in the storage tank, fluid is removed

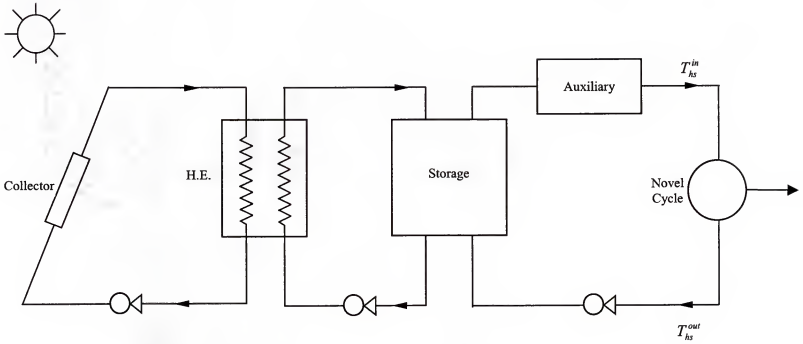


Figure 5.1 A Solar-Driven Ammonia-Based Combined Power/Cooling System

from the bottom of the storage tank. This strategy ensures that the fluid with lowest possible temperature is introduced at the collector inlet for high efficiency. Water enters the novel cycle at a temperature T_{hs}^{in} and leaves at a temperature T_{hs}^{out}. Auxiliary energy is provided by natural gas or other heat source when solar energy is insufficient.

A performance simulation of the solar driven ammonia-based combined power/cooling system was carried out for the month of April in Phoenix, Arizona. The weather data for the month of April in Phoenix, Arizona is listed in Table 5.2.

Table 5.2 Weather Data for the Month of April in Phoenix, Arizona

Location: Phoenix, Arizona Latitude: 33.43°N	Month: April
Horizontal Solar Radiation H_h (kJ/m^2-day)	25560
Horizontal Extraterrestrial Insolation H_h (kJ/m^2-day)	35820
Ambient Temperature T_a (°C)	21

Flat plate collectors with the technical data given in table 5.3 are considered for this application.

Table 5.3 Solar Collector Technical Data

Collector Area A_c (m^2)	3.465
Collector Loss Coefficient $F_R U_c$ (W/m^2-k)	4.13
Collector Optical Efficiency $F_R(\overline{\tau\alpha})$	0.714

For this solar driven ammonia-based combined power/cooling system, our goal is to obtain maximum power/refrigeration output per unit area of solar collectors. Once we have the simulation of the system, optimization program could be used to find the optimum working conditions. The simulation of the system is divided into two parts: the simulation of the novel power/cooling cycle and the simulation of the solar collector system. F-chart method is used here to simulate the solar collector system. It is

incorporated into the optimization program so that the solar driven ammonia-based combined power/cooling system could be optimized based on maximizing power/refrigeration output per unit area of solar collectors.

Optimization Results

In the simulation, 350K water is supplied to the novel power/cooling cycle. The average ambient temperature is 294K in April in Phoenix, Arizona. Optimum working conditions of the novel power/cooling cycle are given in tables 5.4 and 5.5. The state points in first column of the table correspond to the locations given in figure 2.7.

Table 5.4 Optimum Working Conditions

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	299.0	6.3	-84.7	0.2629	0.6778	1.0000
2	299.1	18.6	-83.1	0.2629	0.6778	1.0000
3	338.8	18.6	106.6	0.8574	0.6778	1.0000
4	345.0	18.6	1379.6	4.3655	0.9925	0.1757
5	323.5	18.6	185.5	0.8020	0.9235	0.0151
6	323.5	18.6	1305.9	4.1437	0.9989	0.1606
7	323.5	18.6	1305.9	4.1437	0.9989	0.1606
8	284.6	6.3	1173.1	4.1437	0.9989	0.1606
9	289.0	6.3	1282.8	4.5275	0.9989	0.1606
10	345.0	18.6	108.5	0.9033	0.6164	0.8394
11	304.1	18.6	-82.0	0.3160	0.6164	0.8394
12	304.4	6.3	-82.0	0.3212	0.6164	0.8394

Table 5.5 Cycle Performance Parameters For Conditions In Table 5.4

Boiler Heat Input:	224 kJ/s
Absorber Heat Rejection:	221.9 kJ/s
Turbine Work Output:	21.3 kW
Vapor Quality at Turbine Exit:	91.69 %
Pump Work Input:	1.6 kW
Refrigeration Capacity:	17.6 kW
Total Heat Input:	224 kJ/s
Total Work Output:	19.71 kW
Heat Source Flow Rate:	8.622 kg/s
Heat Source Entrance Temperature:	350 K
Heat Source Exit Temperature:	343.8 K
First Law Efficiency:	16.67 %

It is seen from the tables that 68% ammonia-water mixture is used as basic solution in the absorber/condenser. Absorber/condenser is assumed to operate at 299K, 5K above ambient temperature. Both power and refrigeration are produced from the cycle. Based on 1kg/s basic solution, 19.71 kW power and 17.6 kW (5 tons) refrigeration are produced. It consumes 8.622 kg/s heat source fluid (water), which enters the boiler at 350 K and leaves at 343.8K.

It is also found that when 1500 m^2 solar collector is available, 83% of required heat could be provided by solar energy. That is, solar collector provides 186 kJ/s heat, the additional 38 kJ/s heat has to be provided by other heat sources, such as natural gas, geothermal energy or waste heat. If more solar collectors are available, say, 1800 m^2, then 93% of the required heat could be supplied by solar energy. Figure 5.2 shows the relationship between solar collector area and the fraction of heat supplied by solar energy.

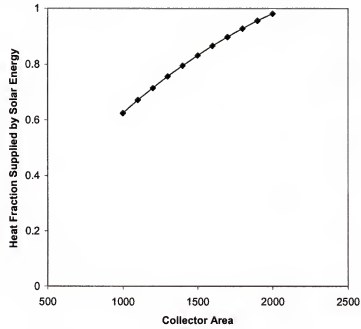


Figure 5.2 Fraction of Heat Supplied by Solar Energy Versus Solar Collector Area

Effect of Water Storage Temperature

Figures 5.3 and 5.4 show the effect of water storage temperature on the system performance. It is seen from figure 5.3 that the first law efficiency of the novel cycle decreases with the water storage temperature. It drops from 20% at 360K to only 8% at 330K. Figure 5.4 shows the work output, refrigeration output and total output produced from the solar driven ammonia-based combined power/cooling system per 1000 m^2 solar collector. In the system, 80% of the heat input to the novel cycle is supplied by solar energy; the other 20% is supplied by other heat sources. It can be seen that work output, refrigeration output and total output all decrease with water storage temperature. Work output drops from 16.2 kW at 360K to 8 kW at 330K. Refrigeration output drops from 15.3 kW at 360K to 4.7 kW at 330K. Total output drops from 31.5 kW at 360K to 12.7 kW at 330K.

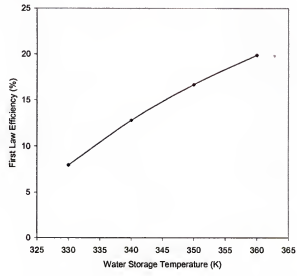


Figure 5.3 Effect of Water Storage Temperature on the First Law Efficiency

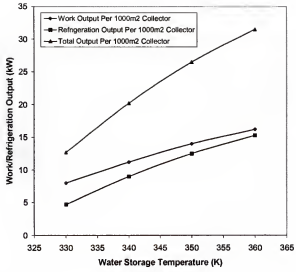


Figure 5.4 Effect of Water Storage Temperature on the Work/Refrigeration Output

Waste Heat

Figure 5.5 is a simplistic representation of the cooling flow diagram for a test reactor which is cooled by 21,000 gpm of 80°F cooling water. When cooling water leaves the reactor, it is heated to 130°F. Under normal operation, the cooling water is cooled in the cooling tower using 50°F, 60psig water from a collection pond.

We can use the novel cycle to recover energy from the cooling water and convert it to power and refrigeration, while reducing the cooling tower requirement. The problem can be described as:

Heat Source Fluid: Water at 40 psig (3.8 bar)

Heat Source Flow Rate: 21,000 GPM (1325 kg/s)

Heat Source Inlet Temperature: 130°F (327K)

Heat Source Outlet Temperature: no restriction, but if higher than 80°F (300K), additional cooling equipment has to be attached.

Cooling Sink: Water at 50°F (283K)

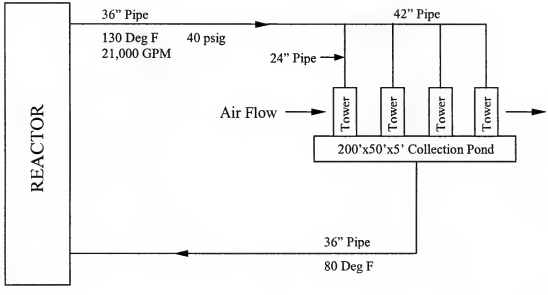


Figure 5.5 Cooling Flow Diagram for a Test Reactor

In order to achieve the best performance, the ammonia-based combined power/cooling cycle is optimized under the above conditions. Absorber is assumed to work at 5K above sink temperature. Tables 5.6 and 5.7 show the optimum working conditions that enable the cycle to produce maximum work output possible. The results are shown based on 1kg/s ammonia basic solution. If we convert it based on 1325 kg/s heat source flow rate, then the total work output from the cycle is 6 MW. The heat source, which is the cooling water for the reactor, leaves the cycle at 307.5K. This temperature is above 300K required by the design. Therefore, additional cooling will be needed before discharging the cooling water. At the optimum working conditions presented in tables 5.6 and 5.7, no refrigeration is produced from the cycle. Tables 5.8

and 5.9 present the optimum working conditions which produce maximum work/refrigeration output. For 1325 kg/s heat source flow rate, the cycle produces 4.5 MW work output along with 622 tons of refrigeration. Again, the cooling water leaves the cycle at a temperature above 300K, which demands additional cooling equipment. However, if we want to avoid using additional cooling equipment, we could select a working condition which is not optimum. One of such conditions is obtained by optimizing the cycle while fixing the leaving temperature of the cooling water at 300K. The results are presented in tables 5.10 and 5.11. At this working condition, the cycle produces 4.8 MW work output while reducing the temperature of the cooling water from 327K (130°F) to 300K (80°F). Thus, no additional cooling facility is required.

Effect of Heat Source Temperature

In the above analysis, the heat source (cooling water for the test nuclear reactor) temperature has been set at 327K, based on an actual reactor. In this section, heat source temperature is varied between 307 K and 347K to study its effect on the performance of the power/cooling cycle while other parameters are unchanged. The optimization results are presented graphically from figures 5.6 to 5.17. Figures 5.6 – 5.11 are based on maximizing cycle work output while figures 5.12 – 5.17 are based on maximizing total cycle work/refrigeration output.

Figure 5.6 shows the amount of heat transferred from heat source (cooling water for the test nuclear reactor) into the power/cooling cycle at optimum conditions based on maximizing cycle work output. Predictably, more heat is transferred from heat source to the cycle at higher heat source temperature. The amount of heat transferred is increased

from 35.1 kJ/s per kg/s heat source fluid at 307 K heat source temperature to 132 kJ/s per kg/s heat source fluid at 347 K heat source temperature.

Figure 5.7 shows the variation of the cycle work output with heat source temperature at optimum conditions based on maximizing cycle work output. Based on 1kg/s heat source fluid, cycle work output is increased from 0.9 kW at 307 K heat source temperature to 10.88 kW at 347 K heat source temperature.

Figure 5.8 shows the variation of absorber pressure and turbine inlet pressure with heat source temperature at optimum conditions based on maximizing cycle work output. It is seen from the figure that the absorber pressure is kept almost constant at different heat source temperatures while turbine inlet pressure increases when heat source temperature increases.

Figure 5.9 shows the variation of the ammonia strong solution concentration in the absorber with heat source temperature at optimum conditions based on maximizing cycle work output. It is seen that the concentration is almost constant. Figure 5.10 shows the variation of the ammonia vapor mass fraction in the boiler/rectifier with heat source temperature. It is seen that ammonia vapor mass fraction increases with heat source temperature.

Figure 5.11 shows the variation of first and second law efficiencies with the heat source temperature at optimum conditions for work output. Both first and second law efficiencies increase when heat source temperature increases. The first law efficiency is increased from 2.56% at 307 K heat source temperature to 8.24% at 347 K heat source temperature. The second law efficiency is increased from 20.78% at 307 K heat source temperature to 40.79% at 347 K heat source temperature.

While figures 5.6 – 5.11 show the optimization results based on maximizing cycle work output, figures 5.12 – 5.17 show the optimization results based on maximizing total cycle work/refrigeration output. In figure 5.12, the amount of heat transferred from the heat source into the cycle at optimum conditions increases when the heat source temperature increases. However, the slope of the curve has a sudden change at 317 K heat source temperature. The reason is that at 307 K heat source temperature, there is no refrigeration found at the optimum conditions even though the optimization objective is to maximize total cycle work/refrigeration output, which in turn causes a shift of optimum working conditions.

Figure 5.13 shows the variation of cycle work output, refrigeration output and total work/refrigeration output with heat source temperature at optimum conditions based on maximizing total cycle work/refrigeration output. It is seen from the figure that work output, refrigeration output and total work/refrigeration output all increase when heat source temperature increases. At 307 K heat source temperature, there is no refrigeration output at optimum conditions. The refrigeration output is increased to 5.15 kW per kg/s heat source fluid at 347 K heat source temperature. Work output is increased from 0.9 kW at 307 K heat source temperature to 8.1 kW at 347 K heat source temperature. Total output is increased from 0.9 kW at 307 K heat source temperature to 13.25 kW at 347 K heat source temperature.

Figure 5.14 shows the variation of absorber pressure and turbine inlet pressure with heat source temperature at optimum conditions based on maximizing total cycle work/refrigeration output. Absorber pressure is almost constant except at 307 K heat source temperature. Due to the shift of optimum working conditions, absorber pressure at

307 K heat source temperature is apparently higher than that at other temperatures. Turbine inlet pressure increases when heat source temperature increases except at 307 K heat source temperature.

Figure 5.15 shows the variation of the ammonia strong solution concentration in the absorber with heat source temperature at optimum conditions based on maximizing total work/refrigeration output. Except at 307 K heat source temperature, the concentration of the ammonia strong solution increases slightly when heat source temperature increases. Figure 5.16 shows the variation of the ammonia vapor mass fraction in the boiler/rectifier with heat source temperature. Except the big shift at 307 K heat source temperature, ammonia vapor mass fraction increases with the heat source temperature.

Figure 5.17 shows the variation of first and second law efficiencies with heat source temperature at optimum conditions based on maximizing total work/refrigeration output. Both first and second law efficiencies increase when heat source temperature increases. The first law efficiency is increased from 2.56% at 307 K heat source temperature to 11.88% at 347 K. The second law efficiency is increased from 20.78% at 307 K heat source temperature to 49.70% at 347 K.

Effect of Sink Temperature

Sink temperature is another significant parameter affecting the performance of the cycle. In this section, sink temperature is varied between 278 K and 293 K to study its effect on the performance of the cycle while other parameters are unchanged. The optimization results are presented graphically in figures 5.18 to 5.29. Figures 5.18 – 5.23

are based on maximizing cycle work output while figures 5.24 – 5.29 are based on maximizing total work/refrigeration output.

Figure 5.18 shows the amount of heat transferred from heat source (cooling water for the test nuclear reactor) into the power/cooling cycle at optimum conditions based on maximizing cycle work output. When sink temperature increases, less heat is transferred from the heat source to the cycle. The amount of heat transferred is decreased from 94.1 kJ/s per kg/s heat source fluid at 278 K sink temperature to 57.8 kJ/s at 293 K.

Figure 5.19 shows the variation of the cycle work output with sink temperature at optimum conditions based on work output. It shows that the work output decreases as the sink temperature increases. Based on 1kg/s heat source fluid, cycle work output is decreased from 5.94 kW at 278 K sink temperature to 2.28 kW at 293 K.

Figure 5.20 shows the variation of the absorber pressure and turbine inlet pressure with the sink temperature at optimum conditions based on work output. When the sink temperature increases, both absorber pressure and turbine inlet pressure increase.

Figure 5.21 shows the variation of the ammonia strong solution concentration in the absorber with the sink temperature at optimum conditions for work output. It shows that the concentration of the ammonia strong solution is almost constant at different sink temperatures. Figure 5.22 shows the variation of the ammonia vapor mass fraction in the boiler/rectifier with sink temperature. It is found that the ammonia vapor mass fraction decreases when the sink temperature increases from 278 K to 293 K.

Figure 5.23 shows the variation of the first and second law efficiencies with the sink temperature at optimum conditions for work output. Both the first and second law efficiencies decrease when the sink temperature increases. The first law efficiency is

decreased from 6.31% at 278 K sink temperature to 3.95% at 293 K. The second law efficiency is decreased from 36.03% at 278 K sink temperature to 28.72% at 293 K.

Figure 5.24 shows the amount of heat transferred from heat source into the power/cooling cycle at optimum conditions for total work/refrigeration output. When the sink temperature increases, less heat is transferred from the heat source to the cycle. The amount of heat transferred is decreased from 78.5 kJ/s per kg/s heat source fluid at 278 K sink temperature to 43.5 kJ/s at 293 K.

Figure 5.25 shows the variation of cycle work output, refrigeration output and total work/refrigeration output with sink temperature at optimum conditions for total work/refrigeration output. It shows that work, refrigeration and total work/refrigeration outputs all decrease when the sink temperature increases. Based on 1kg/s heat source fluid, the work output decreases from 4.54 kW at 278 K sink temperature to 1.64 kW at 293 K. The refrigeration output decreases from 2.48 kW at 278 K sink temperature to 0.5 kW at 293 K. Total work/refrigeration output decreases from 7.02 kW at 278 K sink temperature to 2.14 kW at 293 K.

Figure 5.26 shows the variation of absorber pressure and turbine inlet pressure with sink temperature at optimum conditions for total work/refrigeration output. When the sink temperature increases, both absorber and turbine inlet pressures increase.

Figure 5.27 shows the variation of the ammonia strong solution concentration in the absorber with sink temperature at optimum conditions for total work/refrigeration output. The concentration of the ammonia strong solution varies very little with the sink temperatures. Figure 5.28 shows that the ammonia vapor mass fraction decreases slightly when the sink temperature increases from 278 K to 293 K.

Figure 5.29 shows the variation of first and second law efficiencies with sink temperature at optimum conditions for total work/refrigeration output. Both first and second law efficiencies decrease when the sink temperature increases. The first law efficiency decreases from 8.94% at 278 K sink temperature to 4.92% at 293 K. The second law efficiency decreases from 42.56% to 26.89% in the same range.

Table 5.6 Optimum Working Conditions For Maximizing Work Output Per Unit Mass of Heat Source
(327K heat source temperature, 283K sink temperature)

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	288.0	6.6	2.0	0.2361	0.9076	1.0000
2	288.1	11.3	2.7	0.2361	0.9076	1.0000
3	295.2	11.3	36.3	0.3514	0.9076	1.0000
4	322.0	11.3	1348.2	4.4862	0.9959	0.7481
5	322.0	11.3	9.5	0.5912	0.6453	0.0000
6	322.0	11.3	1348.2	4.4862	0.9959	0.7481
7	322.0	11.3	1348.2	4.4862	0.9959	0.7481
8	293.0	6.6	1275.3	4.4862	0.9959	0.7481
9	293.0	6.6	1275.3	4.4862	0.9959	0.7481
10	322.0	11.3	9.5	0.5912	0.6453	0.2519
11	293.1	11.3	-124.0	0.1569	0.6453	0.2519
12	293.2	6.6	-124.0	0.1590	0.6453	0.2519

Table 5.7 Cycle Performance Parameters For Conditions In Table 5.6

Boiler Heat Input	974.6 kJ/s
Absorber Heat Rejection	920.8 kJ/s
Turbine Work Output	54.5 kW
Vapor Quality at Turbine Exit	98.43 %
Pump Work Input	0.7 kW
Total Heat Input	974.6 kJ/s
Total Work Output	53.81 kW
First Law Efficiency	5.52 %
Heat Source Flow Rate	11.931 kg/s
Heat Source Entrance Temperature	327 K
Heat Source Exit Temperature	307.5 K
Work Output Per Unit Mass of Heat Source Fluid	4.51 kW
Second Law Efficiency	33.98 %

Table 5.8 Optimum Working Conditions For Maximizing Total Output Per Unit Mass of Heat Source
(327K heat source temperature, 283K sink temperature)

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	288.0	4.56	-131.0	0.0953	0.6885	1.0000
2	288.0	8.79	-130.5	0.0953	0.6885	1.0000
3	306.1	8.79	-46.9	0.3768	0.6885	1.0000
4	312.2	8.79	1335.0	4.5544	0.9969	0.1008
5	302.0	8.79	-3.8	0.3848	0.8030	0.0009
6	302.0	8.79	1305.4	4.4576	0.9987	0.0999
7	302.0	8.79	1305.4	4.4576	0.9987	0.0999
8	275.3	4.56	1222.3	4.4576	0.9987	0.0999
9	278.0	4.56	1260.0	4.5940	0.9987	0.0999
10	312.2	8.79	-33.1	0.4513	0.6541	0.9001
11	293.0	8.79	-121.3	0.1598	0.6541	0.9001
12	291.0	4.56	-121.3	0.1617	0.6541	0.9001

Table 5.9 Cycle Performance Parameters For Conditions In Table 5.8

Boiler Heat Input	151.7 kJ/s
Absorber Heat Rejection	147.7 kJ/s
Turbine Work Output	8.3 kW
Vapor Quality at Turbine Exit	96.59 %
Pump Work Input	0.6 kW
Refrigeration Capacity	3.8 kW
Total Heat Input	151.7 kJ/s
Total Work Output	7.75 kW
First Law Efficiency	7.59 %
Heat Source Flow Rate	2.284 kg/s
Heat Source Entrance Temperature	327 K
Heat Source Exit Temperature	311.1 K
Work Output Per Unit Mass of Heat Source Fluid	3.39 kW
Refrigeration Output Per Unit Mass of Heat Source Fluid	1.65 kW
Second Law Efficiency	38.00 %

Table 5.10 Working Conditions With 300 K Fixed Heat Source Exit Temperature (327K heat source temperature, 283K sink temperature)

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	288.0	6.8	23.8	0.2529	0.9376	1.0000
2	288.1	9.2	24.1	0.2529	0.9376	1.0000
3	291.9	9.2	42.2	0.3151	0.9376	1.0000
4	322.0	9.2	1361.1	4.6169	0.9945	0.8633
5	322.0	9.2	-10.6	0.5695	0.5779	0.0000
6	322.0	9.2	1361.1	4.6169	0.9945	0.8633
7	322.0	9.2	1361.1	4.6169	0.9945	0.8633
8	305.5	6.8	1318.7	4.6169	0.9945	0.8633
9	305.5	6.8	1318.7	4.6169	0.9945	0.8633
10	322.0	9.2	-10.6	0.5695	0.5779	0.1367
11	293.1	9.2	-142.6	0.1399	0.5779	0.1367
12	293.1	6.8	-142.6	0.1409	0.5779	0.1367

Table 5.11 Cycle Performance Parameters For Conditions In Table 5.10

Boiler Heat Input:	1131.4 kJ/s
Absorber Heat Rejection:	1095.2 kJ/s
Turbine Work Output:	36.6 kW
Vapor Quality at Turbine Exit:	99.25 %
Pump Work Input:	0.4 kW
Total Heat Input:	1131.4 kJ/s
Total Work Output:	36.2 kW
First Law Efficiency:	3.2 %
Heat Source Flow Rate:	10.007 kg/s
Heat Source Entrance Temperature:	327 K
Heat Source Exit Temperature:	300 K
Work Output Per Unit Mass of Heat Source Fluid	3.62 kW
Second Law Efficiency	27.25 %

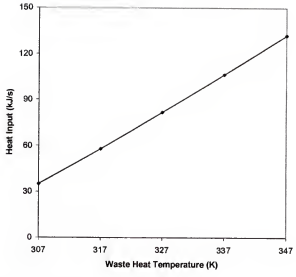


Figure 5.6 Optimum Heat Input per kg/s Waste Heat Fluid at Different Waste Heat Temperatures Based on Maximizing Cycle Work Output

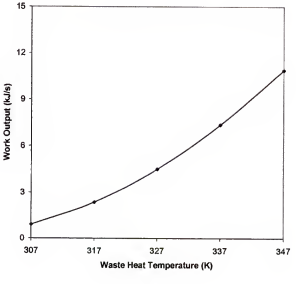


Figure 5.7 Optimum Work Output per kg/s Waste Heat Fluid at Different Waste Heat Temperatures Based on Maximizing Cycle Work Output

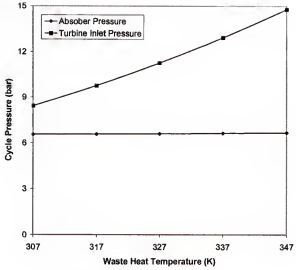


Figure 5.8 Optimum Cycle Pressures at Different Waste Heat Temperatures Based on Maximizing Cycle Work Output

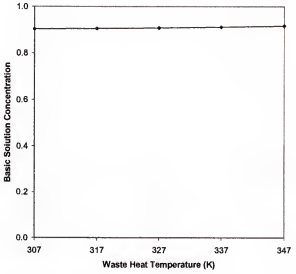


Figure 5.9 Optimum Basic Solution Concentration at Different Waste Heat Temperatures Based on Maximizing Cycle Work Output

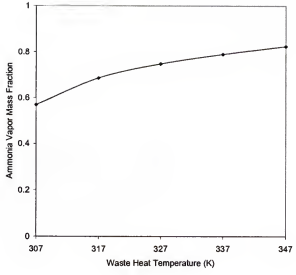


Figure 5.10 Optimum Ammonia Vapor Mass Fraction at Different Waste Heat Temperatures Based on Maximizing Cycle Work Output

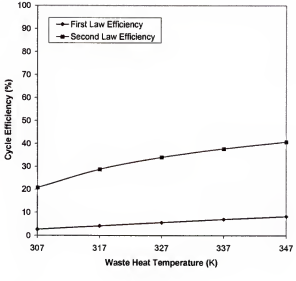


Figure 5.11 Optimum First and Second Law Efficiencies at Different Waste Heat Temperatures Based on Maximizing Cycle Work Output

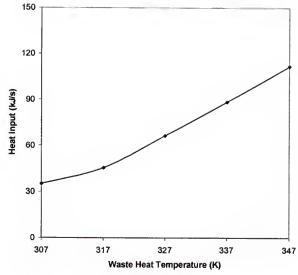


Figure 5.12 Optimum Heat Input per kg/s Waste Heat Fluid at Different Waste Heat Temperatures Based on Maximizing Cycle Total Work/Refrigeration Output

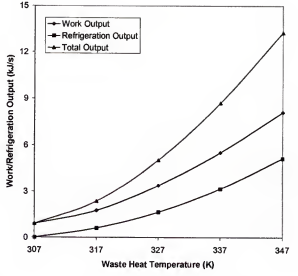


Figure 5.13 Optimum Work/Refrigeration Output per kg/s Waste Heat Fluid at Different Waste Heat Temperatures Based on Maximizing Cycle Total Work/Refrigeration Output

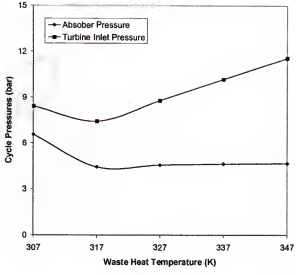


Figure 5.14 Optimum Cycle Pressures at Different Waste Heat Temperatures Based on Maximizing Cycle Total Work/Refrigeration Output

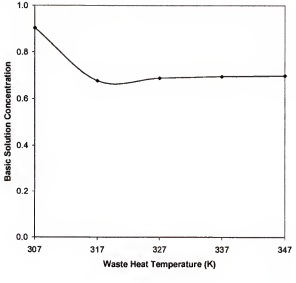


Figure 5.15 Optimum Basic Solution Concentration at Different Waste Heat Temperatures Based on Maximizing Cycle Total Work/Refrigeration Output

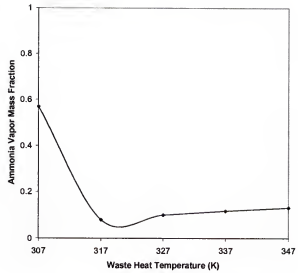


Figure 5.16 Optimum Ammonia Vapor Mass Fraction at Different Waste Heat Temperatures Based on Maximizing Cycle Total Work/Refrigeration Output

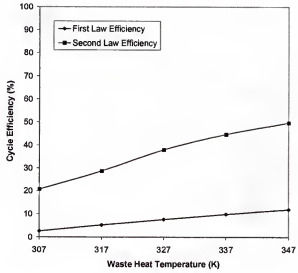


Figure 5.17 Optimum First and Second Law Efficiencies at Different Waste Heat Temperatures Based on Maximizing Cycle Total Work/Refrigeration Output

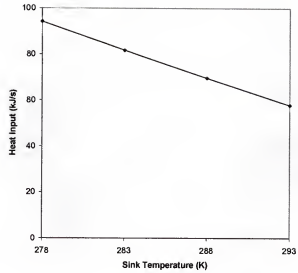


Figure 5.18 Optimum Heat Input per kg/s Waste Heat Fluid at Different Sink Temperatures for Maximum Work Output

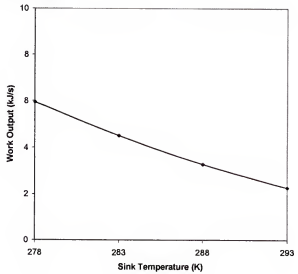


Figure 5.19 Optimum Work Output per kg/s Waste Heat Fluid at Different Sink Temperatures for Maximum Work Output

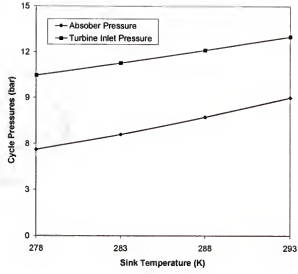


Figure 5.20 Optimum Cycle Pressures at Different Sink Temperatures for Maximum Work Output

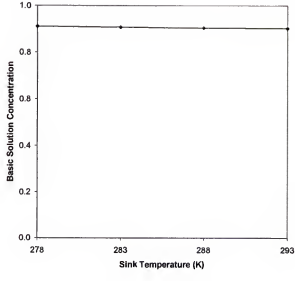


Figure 5.21 Optimum Basic Solution Concentration at Different Sink Temperatures for Maximum Work Output

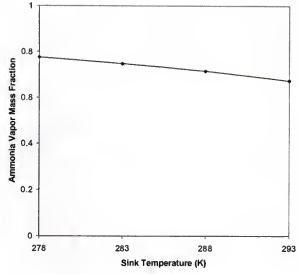


Figure 5.22 Optimum Ammonia Vapor Mass Fraction at Different Sink Temperatures for Maximum Work Output

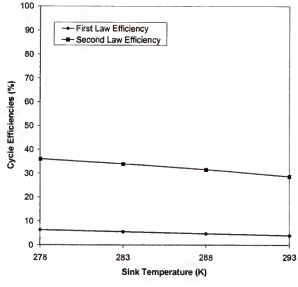


Figure 5.23 Optimum First and Second Law Efficiencies at Different Sink Temperatures for Maximum Work Output

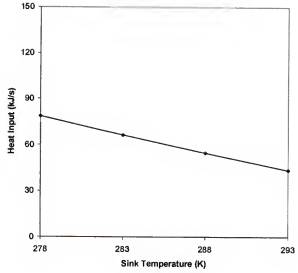


Figure 5.24 Optimum Heat Input per kg/s Waste Heat Fluid at Different Sink Temperatures for Maximum Total Work/Refrigeration Output

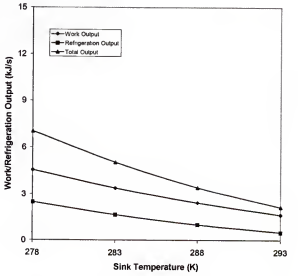


Figure 5.25 Optimum Work/Refrigeration Output per kg/s Waste Heat Fluid at Different Sink Temperatures for Maximum Total Work/Refrigeration Output

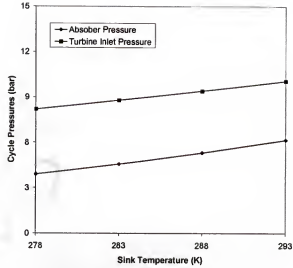


Figure 5.26 Optimum Cycle Pressures at Different Sink Temperatures for Maximum Total Work/Refrigeration Output

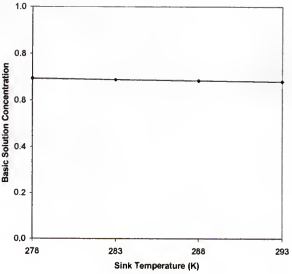


Figure 5.27 Optimum Basic Solution Concentration at Different Sink Temperatures for Maximum Total Work/Refrigeration Output

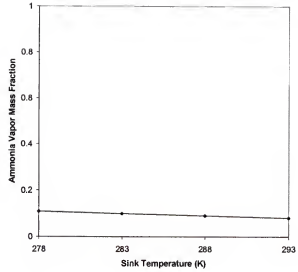


Figure 5.28 Optimum Ammonia Vapor Mass Fraction at Different Sink Temperatures for Maximum Total Work/Refrigeration Output

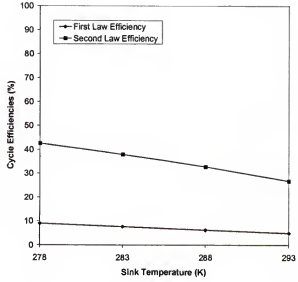


Figure 5.29 Optimum First and Second Law Efficiencies at Different Sink Temperatures for Maximum Total Work/Refrigeration Output

Low Temperature Refrigeration

It is known that in general the refrigeration output from a refrigeration cycle is reduced when the required refrigeration temperature is reduced. In this section, a study is conducted to find out if lower refrigeration temperatures give lower refrigeration output for this cycle and also to see how lower refrigeration temperatures may affect the power output. At each refrigeration temperature, the cycle is optimized for maximum second law efficiency.

Another definition of second law efficiency is introduced in this section. It defines the second law efficiency as the ratio of the useful exergy gained from a system to that supplied to the system. Alefeld (1989), Krakow (1991) and Lee and Sherif (2000) have given detailed discussions on this topic. For this novel power/cooling cycle, it is expressed by Hasan and Goswami (2001) as:

$$\eta_2 = \frac{W_{net} + Q_{cool} / COP_{ideal}}{m_{hs}[(h_{hs}^{in} - h_{hs}^{out}) - T_0(s_{hs}^{in} - s_{hs}^{out})]} \tag{5.1}$$

Where COP_{ideal} is the coefficient of performance for an ideal refrigeration cycle;

h_{hs}^{out} is the outlet enthalpy of the heat source fluid;

s_{hs}^{out} is the outlet entropy of the heat source fluid.

This definition assumes that the spent heat source fluid is reheated in a closed loop and thereby uses the exergy change of the heat source fluid in the denominator. By dividing the refrigeration output by the ideal COP to find its power equivalent in the numerator, this definition emphasizes the importance of the power output over the refrigeration output.

The performance of the ammonia-based combined power/cooling cycle is studied at low refrigeration temperatures. The cycle is optimized for maximum second law efficiency at each refrigeration temperature. Both equations (5.1) and (4.25) are used. To set the refrigeration temperature, a new constraint is added into the existing constraints set listed in chapter 4:

T_8 = fixed value;

The analysis is done for a 360K heat source temperature, which is within the range of flat-plate solar collectors and solar ponds, and 290K as the ambient temperature. Refrigeration temperatures from 265K and below are considered. The simulation starts with a refrigeration temperature of 265K, decreasing it by 10K every time, until no power and refrigeration is produced by the cycle. However, since the thermophysical property program only covers the temperatures down to 230K, uncertainty exists below that temperature.

The optimization results for the cycle at 265K refrigeration temperature based on equation (5.1) are given in tabular form to provide detailed property data at each state point and the energy input and output quantities in the cycle. Table 5.12 shows the optimum working conditions. Table 5.13 gives the cycle performance parameters at the optimum working conditions.

The optimization results based on equation (5.1) are presented graphically in figures 5.30 to 5.34. Figure 5.30 shows that when the refrigeration temperature goes down, both first and second law efficiencies increase slightly at first, and then drop. Both first and second law efficiencies have a maximum at a refrigeration temperature of 245K. The first law efficiency has a maximum of 17.41% and the second law efficiency has a

maximum of 63.7%. The figure also shows that the first and second law efficiencies approach zero at 205K refrigeration temperature.

Table 5.12 Optimum Working Conditions for Heat Source of 360K, Ambient Temperature 290K and Refrigeration Temperature 265K

Point	T(K)	P(bar)	h(kJ/kg)	s(kJ/kg.K)	X	Flow Rate (kg/s)
1	295.0	0.439	-56.8	0.2990	0.2253	1.0000
2	295.0	2.759	-56.6	0.2990	0.2253	1.0000
3	347.0	2.759	200.0	1.0910	0.2253	1.0000
4	355.0	2.759	1666.9	5.9235	0.8232	0.0779
5	331.2	2.759	65.7	0.7568	0.2887	0.0159
6	331.2	2.759	1455.8	5.4536	0.9598	0.0621
7	331.2	2.759	1455.8	5.4536	0.9598	0.0621
8	265.0	0.439	1199.2	5.4536	0.9598	0.0621
9	285.0	0.439	1310.2	5.8529	0.9598	0.0621
10	355.0	2.759	226.9	1.1114	0.1766	0.9379
11	300.0	2.759	-5.6	0.3998	0.1766	0.9379
12	300.1	0.439	-5.6	0.4006	0.1766	0.9379

Table 5.13 Cycle Performance Parameters For Conditions In Table 5.12

Boiler Heat Input:	141.7 kJ/s
Absorber Heat Rejection:	132.9 kJ/s
Turbine Work Output:	15.9 kW
Vapor Quality at Turbine Exit:	94.33 %
Pump Work Input:	0.3 kW
Refrigeration Capacity:	6.9 kW
Total Heat Input:	141.7 kJ/s
Total Work Output:	15.68 kW
First Law Efficiency:	15.93 %
Second law efficiency:	62.18 %

Figure 5.31 shows the variation of the absorber and turbine inlet pressures with refrigeration temperature. When the refrigeration temperature drops from 265K to 205K, both the absorber and the turbine inlet pressures first increase and then decrease below 245K refrigeration temperature.

Figure 5.32 shows that the concentration of the ammonia solution in the absorber increases at first as the refrigeration temperature decreases, and then decreases. Figure

5.33 shows that the ammonia vapor fraction increases slightly as the refrigeration temperature drops from 265K to 245K, and then decreases for refrigeration temperature below 245K.

Figure 5.34 shows the variation of normalized work output and refrigeration output with refrigeration temperature. Generally, normalized work and refrigeration outputs increase with the refrigeration temperature. However, COP of the ideal refrigeration cycle has higher values at higher refrigeration temperatures. Therefore, when the refrigeration temperature is above 245K, the ideal COP is so large that the contribution of the refrigeration output to the second law efficiency becomes very small. Consequently, optimization reduces the refrigeration output to obtain a slight increase in the work output. Therefore, refrigeration output starts to drop when the refrigeration temperature goes above 245K.

Since refrigeration is the main intended output in this study, the cycle was also optimized for the second law efficiency in equation (4.25) where refrigeration is given a weight equal to the power output. The optimization results based on equation (4.25) are presented graphically from figures 5.35 to 5.39. Figure 5.35 shows the variation of the second law efficiency with refrigeration temperature. Unlike the results shown in Fig. 5.30, the second law thermal efficiency of the cycle based on equation (4.25) always decreases as refrigeration temperature goes down. At 265K, the cycle has a second law thermal efficiency of 52.2%, and it decreases as the refrigeration temperature goes down. It approaches zero at 205K refrigeration temperature. The first law efficiency of the cycle also decreases with the refrigeration temperature.

Figure 5.36 shows the variation of the absorber and turbine inlet pressures with refrigeration temperature. Starting at 265K, the absorber pressure decreases with refrigeration temperature while it shows a peak in Fig. 5.31. For lower refrigeration temperature, in order to maintain the quality level of the ammonia vapor at the exit of the turbine, the exhaust pressure of the turbine has to be lowered correspondingly. Under idealized conditions, the absorber pressure is equal to the turbine exhaust pressure and therefore is lower at low refrigeration temperatures. The turbine inlet pressure also decreases with refrigeration temperature. When the concentration of ammonia basic solution gets lower at a low refrigeration temperature, in order to produce enough ammonia vapor in the boiler, the boiler pressure has to go down correspondingly. Since turbine inlet pressure is the same as the boiler pressure under idealized conditions, it goes down simultaneously.

Figure 5.37 shows a variation of the ammonia solution concentration in the absorber with the refrigeration temperature. Compared with Fig. 5.32, it is found that the optimal basic solution concentration based on equation (4.25) has no peak. It decreases when the refrigeration temperature decreases. In order to generate as much ammonia vapor in the boiler as possible, a saturation state for ammonia solution is desired in the absorber. For saturated ammonia solution, its concentration is determined by its temperature and pressure. When the temperature is lower or the pressure is higher, the concentration of the saturated ammonia solution is higher. However, the temperature of the absorber is bounded by the ambient temperature. In this analysis, 5K above the ambient temperature is chosen for the absorber. So the concentration of the ammonia basic solution is only decided by the absorber pressure. As the absorber pressure

decreases with the refrigeration temperature, the concentration of the ammonia solution in the absorber also decreases. At 205K refrigeration temperature, the concentration of the basic solution at the optimum conditions is only 6.8%.

Even though the boiler pressure goes down with the refrigeration temperature, the ammonia vapor generated in the boiler is very little at very low refrigeration temperatures due to the low concentration of the feeding ammonia solution. This point becomes clear from Fig. 5.38. The vapor fraction, which is the ratio of the mass flow rate of the ammonia vapor at point 6 to that of the ammonia basic solution at point 1, is almost zero at 205K refrigeration temperature. However, in Fig. 5.33, the vapor fraction reaches the maximum at 245K refrigeration temperature, where the concentration of the ammonia solution in the absorber is also the highest.

Figure 5.39 shows that the work and refrigeration outputs (per kg/s heat source fluid) decrease with the refrigeration temperature. It is understandable that with lower vapor flow through the turbine, lower amount of work and refrigeration will be produced. No peak appears for refrigeration output as in Fig. 5.34.

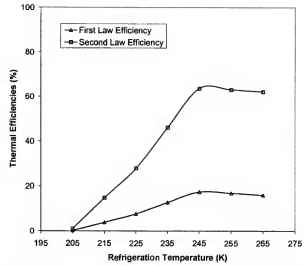


Figure 5.30 Optimum First and Second Law Efficiencies at Different Refrigeration Temperatures Based on Equation (5.1)

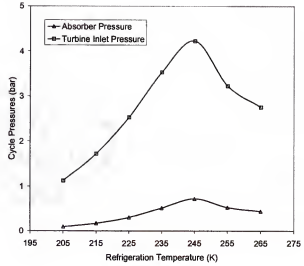


Figure 5.31 Optimum Cycle Pressures at Different Refrigeration Temperatures Based on Equation (5.1)

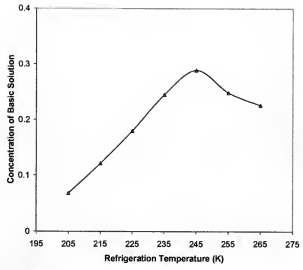


Figure 5.32 Optimum Concentration of Basic Solution at Different Refrigeration Temperatures Based on Equation (5.1)

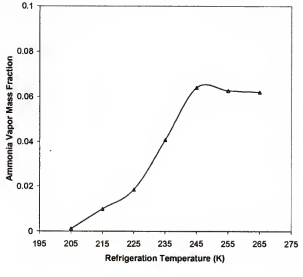


Figure 5.33 Optimum Ammonia Vapor Mass Fraction at Different Refrigeration Temperatures Based on Equation (5.1)

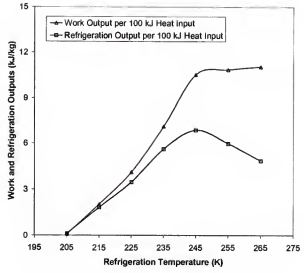


Figure 5.34 Optimum Work and Refrigeration Outputs at Different Refrigeration Temperatures Based on Equation (5.1)

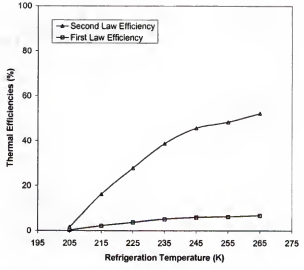


Figure 5.35 Optimum First and Second Law Efficiencies at Different Refrigeration Temperatures Based on Equation (4.25)

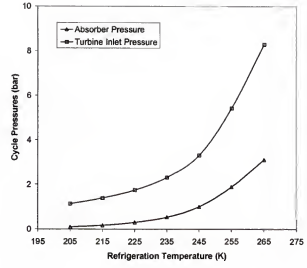


Figure 5.36 Optimum Cycle Pressures at Different Refrigeration Temperatures Based on Equation (4.25)

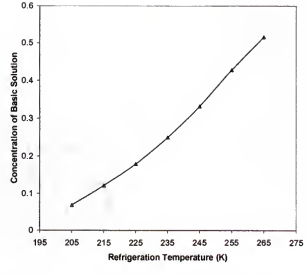


Figure 5.37 Optimum Concentration of Basic Solution at Different Refrigeration Temperatures Based on Equation (4.25)

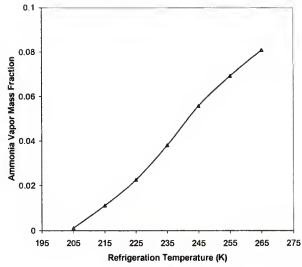


Figure 5.38 Optimum Ammonia Vapor Mass Fraction at Different Refrigeration Temperatures Based on Equation (4.25)

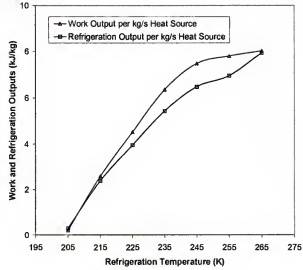


Figure 5.39 Optimum Work and Refrigeration Outputs at Different Refrigeration Temperatures Based on Equation (4.25)

CHAPTER 6
CONCLUSIONS

The Rankine cycle and Brayton cycle are the two most successful thermodynamic cycles ever invented. They have been used in the power generation and other industries since nineteenth century. Although science and technology have gone through rapid renovation since then, no new cycle has been invented to replace their dominant positions. Recently, due to the demand to further improve thermodynamic efficiency of the power plant and find a new, innovative way to utilize renewable energy resources, such as geothermal energy and solar energy, intensive research is being done to find new, effective thermodynamic cycles. Development of the Kalina cycle is one such result. However, its advantage is compromised by its extremely complicated configuration. More research still needs to be done to find a satisfactory thermodynamic cycle. In this dissertation, a novel cycle, ammonia-based combined power/cooling cycle, suggested by Goswami (1995) is investigated and has been found to be suitable for many low temperature power conversion applications.

For this ammonia-based combined power/cooling cycle, a parametric analysis was conducted under idealized conditions (Simulation program was later modified to include irreversibilities to yield a more realistic study. The effect of each irreversibility factor as well as their combined effect on the cycle performance has been carefully studied.) The parameters studied include turbine inlet pressure, boiler temperature, rectifier

temperature, superheater temperature, absorber temperature and pressure. The parametric analysis was conducted within the following ranges of the parameters:

Turbine inlet pressure: 18 - 32 bar

Boiler temperature: 390 - 420 K

Rectifier temperature: 350 - 400K

Superheater temperature: 400 - 500K

Absorber temperature: 280 - 310K

Absorber pressure: 1 - 3bar

Through the detailed parametric analysis, it was seen that the cycle conditions could be optimized for maximum performance.

The new thermodynamic cycle was optimized using the Generalized Reduced Gradient (GRG) algorithm for the objective function written as:

$$\eta = f(T_{\text{absorber}}, T_{\text{boiler}}, T_{\text{condenser}}, T_{\text{superheater}}, P_{\text{high}}, P_{\text{low}}, T_{hs}^{in}, T_{hs}^{out})$$

Second law thermal efficiency, which is the true measure of the efficiency of resource utilization, was chosen as our primary optimization objective.

Two typical heat source temperatures, 360 K and 440K, were studied. A heat source temperature of 360K is within the range of flat-plate solar collectors and solar ponds while a heat source temperature of 440K is within the range of some geothermal sources, and solar resources using CPC or other low concentration solar collectors. It was found that for a source temperature of 360K, both power and refrigeration outputs are achieved under optimum conditions. On the other hand, for a source temperature of 440K, optimum conditions do not provide any refrigeration. However, refrigeration can be obtained even for this temperature under non-optimum performance conditions.

Although second law efficiency was chosen as the primary optimization objective, the cycle may be optimized for any desired performance parameter. Examples are provided for 360 K heat source temperature based on maximizing work output and refrigeration output per unit mass of heat source fluid, respectively. A comparison of three optimization results shows that optimum conditions for maximum work output per unit mass of heat source fluid produces no refrigeration though more work output. Optimum conditions for maximum refrigeration output per unit mass of heat source fluid are close to those for maximum second law efficiency.

The effect of ambient temperature on optimum cycle performance was investigated over the range of 280K to 310K. It was found that for a source temperature of 360K, all performance parameters, including first and second law efficiencies, power and refrigeration output decrease as the ambient temperature goes up. On the other hand, for a source temperature of 440K, the second law efficiency varies very little as the ambient temperature goes up from 280 to 310K. However, work output decreases because the exergy of the source fluid goes down as the ambient temperature goes up.

Some applications of this novel power/cooling cycle were also studied. One application is to use low cost flat-plate solar collectors as heat source for this novel power/cooling cycle. A system performance simulation of a solar driven ammonia-based combined power/cooling system was carried out for the month of April in Phoenix, Arizona. It uses f-chart method to simulate solar collector system. The optimization of the solar driven ammonia-based combined power/cooling system was done for maximizing the combined power and refrigeration outputs per unit area of solar collectors.

A second application of the novel power/cooling cycle for utilizing the waste heat from a test nuclear reactor was also analyzed. Though the temperature of the cooling water is only 130°F (327K), useful energy could be recovered from it by this novel power/cooling cycle. The cycle was optimized for work output, total work/refrigeration output, and for work output while reducing the cooling water temperature to a desired value. It was found that for 21,000 GPM cooling water, 6 MW work output or 4.5 MW work output plus 622 tons refrigeration could be generated.

The effect of the waste heat temperature was investigated over the range of 307K to 347K. It was found that all performance parameters, including first and second law efficiencies, power and refrigeration output increase as the temperature of the waste heat fluid goes up. The effect of the sink temperature was also investigated. The study finds that the first and second law efficiencies, and power and refrigeration outputs all decrease when the sink temperature goes up.

The third application analyzed in this study was for low temperature refrigeration. The performance of the cycle was studied at low refrigeration temperature. It was found that a refrigeration temperature as low as 205K could be achieved. However, the cycle performance generally worsens when the refrigeration temperature decreases. Both first and second law efficiencies therefore drop as the refrigeration temperature goes down. However, for one definition of second law efficiency, where the reciprocal of an ideal coefficient of performance is used as a weight factor for the refrigeration output, the first and second law efficiencies increase slightly as the refrigeration temperature decreases and then decrease, reaching maxima at 245K refrigeration temperature.

An experiment system is being set up to demonstrate the practicability of the ammonia-based combined power/cooling cycle. In addition, using other multi-component working fluids instead of ammonia-water mixtures will be investigated.

APPENDIX
CYCLE SIMULATION PROGRAM WITH OPTIMIZATION

The optimization program consists of the following files:

1. awpcc.c
2. conbsetc.c
3. datain.c
4. diretc.c
5. grgcheck.c
6. grgglobl.c
7. grgitn.c
8. grgmem.c
9. grgsub.c
10. initlz.c
11. main.c
12. outretc.c
13. phetc.c
14. property.c
15. userval.c
16. awpcc.h [header file for awpcc.c]
17. grgcodes.h [contains grg2 return codes and their meanings]
18. grgglobl.h [header file for grg2 system globals]
19. property.h [header file for property.c]

To limit pages, only main.c, awpcc.c, awpcc.h, property.c, and property.h are listed here. main.c is the main calling program in which the initial values of the free variables and their upper and lower bounds are specified. awpcc.c is the file which does the thermodynamic calculations of the ammonia-based combined power/cooling cycle. It contains subroutine gcomp and another subroutine called by gcomp. awpcc.h is its header file. property.c does property calculation of ammonia/water mixture. property.h is its header file.

main.c

```
/*****************************************************************************/
/* THIS PROGRAM IS A MAIN CALLING PROGRAM FOR A PROGRAM GROUP.               */
/* THE PROGRAM GROUP OPTIMIZES THE PERFORMANCE OF AN AMMONIA-BASED           */
/* COMBINED POWER/COOLING CYCLE BY GRGC USING THE SUBROUTINE                 */
/* INTERFACE VERSION 2.0 BETA                                                */
/*****************************************************************************/

#define MAXVARS 16
#define MAXROWS 16

#include <stdio.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>

#include "grgcodes.h"  /* contains symbolic termination codes */

double iipow(double,int);

void  grgsetparameter(char *, long, double);
char *grg2_get_terminationmsg(void);

int  grgsub(int nvars, int nrows, int nobj, char *title, int inprint,
                        int outprint, char *var[],char *con[],
                        double xlb[],double xub[],double glb[],double gub[],
                        double g[],double xx[],double rmult[],double redgr[],
                        int inbind[], int nonbas[],int *nnonb,int *nbind,
                        FILE *ioout);

void main()
{
        /*===========================================================================*/
        /*   Main program for GRG using subroutine interface.          */
        /*                                                             */
        /*   Note on GRG2 Output:                                      */
        /*    Default Output Destination is stdout                     */
        /*    Assigning ioout=NULL is equivalent to ioout=stdout       */
        /*    Assigning ioout to an open FILE * will cause all output  */
        /*    to be sent to that file (see notes below)                */
        /*                                                             */
        /*    To quiet all output from GRG2, do the following:         */
        /*     (1) set inprint=0                                       */
        /*     (2) set outprint=0                                      */
        /*     (3) set ipr=0 with the  following function call         */
        /*        grgsetparameter("ipr",(long)0,(double)0.0);          */
        /*                                                             */
        /*  To See an iteration log on the screen when output is       */
        /*  being sent to a file (ipr must be > 0), make the following */
        /*  call and an iteration log will be written to stdout        */
        /*    grgsetparameter("ioterm",(long)1,(double)0.0);           */
```

```
        /*   Be default, no screen output is provided,                */
        /*                                                            */
        /*                                                            */
        /*============================================================================*/

/* ------ grgsub arguments -----*/
 char  title[81];
 char *var[MAXVARS], *con[MAXVARS];

 double  xlb[MAXVARS],xub[MAXVARS],glb[MAXROWS],
                gub[MAXROWS],g[MAXROWS],xx[MAXVARS],rmult[MAXROWS],
                redgr[MAXVARS];

 int    inbind[MAXROWS],nonbas[MAXVARS],nbind,nnonb,info,
                nvars,nrows,inprint,outprint,nobj;
 FILE *ioout;
/* ----- end grgsub arguments ------*/

char    filename[31];

int i;

/******************** MAKE CHANGES BELOW **************************/


/*  IF YOU WANT REQUESTED OUTPUT TO GO TO A FILE YOU SPECIFY WHEN */
/*  YOU EXECUTE THIS PROGRAM THEN REMOVE THE COMMENTS SURROUNDING */
/*  THE CODE SEGMENT BELOW WHICH ASKS FOR A FILE PATH AND NAME AND*/
/*  ASSIGNS THAT TO THE FILE IDENTIFIER ioout, OVERRIDING stdout. */

/*  OPENING COMMENT INDICATOR FOR FILENAME INPUT FOLLOWS ON LINE BELOW */

        strcpy(filename, "");
        do
        {
                        printf("\nEnter output data file path and name: ");
                        scanf ("%30s", filename);
                        if ((ioout = fopen(filename, "w") ) == NULL)
                                {printf("Could not open file: %s. Please retry.\n",filename);
                                strcpy(filename, ""); }
                        else
                                printf("File %s opened for output.\n",filename);
        }
        while (strcmp(filename, "") == 0 );

/*  CLOSING COMMENT INDICATOR FOR FILENAME INPUT ON LINE ABOVE  */

/*************************************************************************/
/*  ..... required scalar inputs to grgsub .....  */
/*************************************************************************/

/*    nvars -- NUMBER OF VARIABLES                          */

nvars = 8;

/*    nrows -- NUMBER OF FUNCTIONS INCLUDING OBJECTIVE          */
```

```
nrows = 13;

/*    nobj  -- INDEX OF COMPONENT OF VECTOR {G} IN  SUBROUTINE   */
/*             GCOMP CORRESPONDING TO OBJECTIVE FUNCTION          */

nobj = 13;


/*  ioout -- FILE* indicates destination of grg2 output           */
/*           ioout=NULL will default output to stdout             */
/*           Quiet all output by setting inprint=0,outprint=0  and */
/*  calling  grgsetparameter("ipr",(long) 1,(double) 0.0)         */

/* ioout = stdout;  */

/*  inprint  -- 0 = no echo of initial point , function values    */
/*                and settings                                    */

inprint = 1;

/* outprint  -- 0 = no output of final point , function values    */
/*              and run statistics                                */

outprint = 1;


/*    title CHARACTER STRING OF AT MOST 80 CHARACTERS USED TO     */
/*          IDENTIFY THE PROBLEM IN ANY PRINTED REPORTS.         */

strcpy ( title, "Ammonia/Water Based Power Cycle Optimization." );

/*********************************************************************/
/*        allocate memory for row and column names                   */
/*  If you do not want to supply row and column names,  set          */
/*   var[1]=NULL and con[1]=NULL  and grg2 will leave all names      */
/*   blank                                                           */
/*********************************************************************/
  for(i = 1; i <= nvars; ++i) {
                    var[i] = (char *) malloc((size_t) 11*sizeof(char));
                    if(var[i]==NULL) {
                              printf("\n.. Unable to allocate space for variable names");
                              return;
                    }
  }
  for(i = 1; i <= nrows; ++i) {
                    con[i] = (char *) malloc((size_t) 11*sizeof(char));
                    if(con[i]==NULL) {
                              printf("\n.. Unable to allocate space for constraint names");
                              return;
                    }
  }
/*************************************************************************/
/* var - char * ARRAY WITH MAX SUBSCRIPT {nvars}.  CONTAINS CHARACTER  */
/*           STRING NAMES FOR THE VARIABLES. UP TO 10 CHARACTERS.    */
/*************************************************************************/
```

```
  strcpy ( var[1], "t_ab" );
  strcpy ( var[2], "t_boiler" );
  strcpy ( var[3], "t_cond" );
  strcpy ( var[4], "t_super" );
  strcpy ( var[5], "p_high" );
  strcpy ( var[6], "x_com" );
  strcpy ( var[7], "ths1" );
  strcpy ( var[8], "ths2" );

/* con - char * ARRAY WITH MAX SUBSCRIPT {nrows}. CONTAINS CHARACTER   */
/*          STRING NAMES FOR THE FUNCTIONS. UP TO 10 CHARACTERS.   */

  strcpy ( con[1], "row1   ge" );
  strcpy ( con[2], "row2   ge" );
  strcpy ( con[3], "row3   ge" );
  strcpy ( con[4], "row4   ge" );
  strcpy ( con[5], "row5   ge" );
  strcpy ( con[6], "row6   ge" );
  strcpy ( con[7], "row7   ge" );
  strcpy ( con[8], "row8   ge" );
  strcpy ( con[9], "row9   ge" );
  strcpy ( con[10],"row10  ge" );
  strcpy ( con[11],"row11  ge" );
  strcpy ( con[12],"row12  ge" );
  strcpy ( con[13],"row13 obj" );

/*************************************************************************/
/*  ..... array inputs to grgsub  .....  */
/*************************************************************************/
/*  xx - DOUBLE ARRAY WITH MAX SUBSCRIPT {nvars}.CONTAINS INITIAL */
/*       VARIABLE VALUES. VALUES WHICH DO NOT SATISFY THE GIVEN */
/*       VARIABLE BOUNDS WILL BE CHANGED TO THE BOUND NEAREST   */
/*       THE VALUE. Final Variable values are returned in xx    */

xx[1] = 295.   ;
xx[2] = 380.   ;
xx[3] = 380.   ;
xx[4] = 435.   ;
xx[5] =  30.   ;
xx[6] =  0.7   ;
xx[7] = 440.   ;
xx[8] = 330.   ;

/*   xlb - DOUBLE ARRAY CONTAINING LOWER BOUNDS FOR VARIABLES.      */
/*       VARIABLE LOWER BOUNDS ARE IN POSITIONS FROM 1 TO {nvars}. */
/*       IF VARIABLE i HAS NO LOWER BOUND, SET xlb[i] TO -1.0e30   */
/*   xub - DOUBLE ARRAY CONTAINING UPPER BOUNDS FOR VARIABLES.      */
/*       VARIABLE LOWER BOUNDS ARE IN POSITIONS FROM 1 TO {nvars}. */
/*       IF NO UPPER BOUND, SET xub[i] TO 1.0e30                    */

xlb[1] =  295.    ; xub[1] =  295.   ;
xlb[2] =  295.    ; xub[2] =  440.   ;
xlb[3] =  295.    ; xub[3] =  440.   ;
xlb[4] =  295.    ; xub[4] =  440.   ;
xlb[5] =   5.     ; xub[5] =   80.   ;
```

```
xlb[6] =  0.1    ; xub[6] =  0.95  ;
xlb[7] = 440.    ; xub[7] =  440.  ;
xlb[8] = 330.    ; xub[8] =  440.  ;

/*   glb - DOUBLE ARRAY CONTAINING LOWER BOUNDS FOR FUNCTIONS.      */
/*         FUNCTION LOWER BOUNDS ARE IN POSITIONS FROM 1 TO {nrows}.*/
/*         IF A FUNCTION HAS NO LOWER BOUND, SET xlb[i] TO -1.0e30  */
/*   gub - DOUBLE ARRAY CONTAINING UPPER BOUNDS FOR FUNCTIONS.      */
/*         FUNCTION INDEXES ARE THE SAME AS ABOVE FOR glb.          */
/*         IF NO UPPER BOUND, SET gub[i] TO 1.0e30                  */
/*                                                                  */
/*   NOTE 1: IT DOES NOT MATTER WHAT YOU USE FOR THE BOUNDS OF      */
/*         THE OBJECTIVE FUNCTION IN {glb} AND {gub}.               */
/*   NOTE 2: IF YOU WISH TO FIX A VARIABLE AT A CERTAIN VALUE AND   */
/*         HAVE GRGSUB LEAVE IT UNCHANGED, SET ITS ENTRIES IN xlb */
/*         AND xub TO THAT VALUE                                    */
/*   NOTE 3: IF g[i] IS AN EQUALITY CONSTRAINT, EQUAL TO, SAY, B,   */
/*         SET glb[i] = gub[i] = B                                  */
/*   NOTE 4: IF FUNCTION g[i] IS TO BE IGNORED IN THE CURRENT RUN   */
/*         OF GRGSUB, SET glb[i] TO -1.0e30 AND gub[i] TO 1.0e30  */

glb[1] = -1.0e+30  ; gub[1] =  0.00    ;
glb[2] = -1.0e+30  ; gub[2] =  0.00    ;
glb[3] = -1.0e+30  ; gub[3] = -0.10    ;
glb[4] =  0.00     ; gub[4] =  100.    ;
glb[5] = -1.0e+30  ; gub[5] = -0.10    ;
glb[6] =  5.       ; gub[6] =  1.0e+30 ;
glb[7] =  90.      ; gub[7] =  100.    ;
glb[8] =  0.00     ; gub[8] =  100.    ;
glb[9] =  5.       ; gub[9] =  1.0e+30 ;
glb[10]=  5.       ; gub[10]=  1.0e+30 ;
glb[11]=  5.       ; gub[11]=  1.0e+30 ;
glb[12]=  3.       ; gub[12]=  1.0e+30 ;
glb[13]=  0.00     ; gub[13]=  100.    ;

/*  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  */
/*  GRG PARAMETERS WITH DEFAULT VALUES                               */
/*  You may change any of these (or none) before calling GRGSUB      */
/*                                                                   */
/*  Changes to the default value of any parameter are made by        */
/*  calling function grg2setparameter(string,ivalue,dvalue)          */
/*   where 'string' is the parameter name from the list below,       */
/*   lvalue is an long int value for integer parameters,             */
/*    (some int values may be too large for 16 bit ints) and         */
/*   dvalue is a double value for double parameters                  */
/*   Examples are given for  epnewt  and  ipr                        */
/*  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  */
/*   VARIABLE    DEFAULT                                             */
/*    NAME        VALUE        DESCRIPTION                          */
/*                                                                   */
/*  maxb  -- UPPER LIMIT ON NUMBER OF BINDING CONSTRAINTS.           */
/*         USE {NROWS} <default> IF UNSURE OF A SMALLER LIMIT        */
/*                                                                   */
/*  maxr  -- MAXIMUM ALLOWABLE SIZE OF APPROXIMATE                   */
/*         HESSIAN - USE default {NVARS} IF YOU WANT A               */
/*         QUASI-NEWTON METHOD TO BE USED AT EVERY ITERATION         */
```

```
/*          (FASTEST METHOD IF NOT TOO MANY VARIABLES)          */


/*  EPSILONS */
/*    1  epnewt---1.0E-06--- A CONSTRAINT IS ASSUMED TO BE      */
/*               BINDING IF IT IS WITHIN THIS EPSILON           */
/*               OF ONE OF ITS BOUNDS.                          */
/*                                                              */

/*     grgsetparameter("epnewt",(long)0,(double)1.0e-10);       */

/*    2  epinit---1.0E-06--- IF IT IS DESIRED TO RUN THE        */
/*               PROBLEM WITH {epnewt} INITIALLY SET FAIRLY     */
/*               LARGE AND THEN TIGHTENED AT THE END OF THE     */
/*               OPTIMIZATION THEN THIS IS ACCOMPLISHED BY      */
/*               ASSIGNING {epinit} THE INITIAL TOLERANCE       */
/*               AND {epnewt} THE FINAL ONE.                    */


/*    3  epstop---1.0E-04--- IF THE FRACTIONAL CHANGE IN THE    */
/*               OBJECTIVE IS LESS THAN {epstop} FOR {nstop}    */
/*               CONSECUTIVE ITERATIONS, THE PROGRAM WILL       */
/*               STOP. PROGRAM WILL ALSO STOP IF KUHN-TUCKER    */
/*               OPTIMALITY CONDITIONS ARE SATISFIED TO WITHIN  */
/*               {epstop}.                                      */

          grgsetparameter("epstop",(long)0,(double)1.0e-14);

/*    4  epspiv---10.0E-3--- IF, IN CONSTRUCTING THE BASIS      */
/*               INVERSE, THE ABSOLUTE VALUE OF A PROSPECTIVE   */
/*               PIVOT ELEMENT IS LESS THAN {epspiv}, THE       */
/*               PIVOT WILL BE REJECTED AND ANOTHER PIVOT       */
/*               ELEMENT WILL BE SOUGHT.                        */

          grgsetparameter("epspiv",(long)0,(double)1.0e-5);

/*    5  ph1eps--- 0.0  --- IF NONZERO, THE PHASE 1 OBJECTIVE    */
/*               IS AUGMENTED BY A MULTIPLE OF THE TRUE         */
/*               OBJECTIVE.  THE MULTIPLE IS SELECTED SO THAT,  */
/*               AT THE INITIAL POINT, THE RATIO OF THE TRUE    */
/*               OBJECTIVE AND SUM OF THE INFEASIBILITIES IS    */
/*               {ph1eps}.                                      */



/*    6  pstep --- 1.0E-4  --THIS IS THE STEP SIZE USED IN PARSH */
/*                  AND PARSHC FOR ESTIMATING PARTIAL           */
/*                  PARTIAL DERIVATIVES OF THE FUNCTIONS        */
/*                  WITH RESPECT TO THE VARIABLES.              */

          grgsetparameter("pstep",(long)0,(double)1.0e-3);

/* LIMITS */
/*    1  nstop --- 3    --- IF THE FRACTIONAL CHANGE IN THE      */
/*               OBJECTIVE IS LESS THAN {epstop} FOR {nstop}    */
/*               CONSECUTIVE ITERATIONS, THE PROGRAM WILL       */
```

```
/*              STOP.                                   */

        grgsetparameter("nstop",(long)6 , (double)0.0);

/*   2  itlim --- 10    --- IF SUBROUTINE NEWTON TAKES          */
/*              {itlim} ITERATIONS WITHOUT CONVERGING          */
/*              SATISFACTORILY, THE ITERATIONS ARE STOPPED     */
/*              AND CORRECTIVE ACTION IS TAKEN.                */



/*   3  limser---1,000 --- IF THE NUMBER OF COMPLETED ONE      */
/*              DIMENSIONAL SEARCHES EQUALS {limser},          */
/*              OPTIMIZATION WILL TERMINATE.                   */

        grgsetparameter("limser",(long)6000,(double)0.0);



/* PRINT CONTROL */
/*   1  ipr  -- 0  - SUPPRESS ALL OUTPUT PRINTING EXCEPT       */
/*                   INITIAL AND FINAL REPORTS.               */
/*    (default) ==> - 1 - PRINT ONE LINE OF OUTPUT FOR EACH ONE  */
/*                   DIMENSIONAL SEARCH.                      */
/*                - 2 - PROVIDE MORE DETAILED INFORMATION ON   */
/*                   THE PROGRESS OF EACH ONE DIMENSIONAL     */
/*                   SEARCH.                                  */
/*                - 3 - EXPAND THE OUTPUT TO INCLUDE THE PROBLEM */
/*                   FUNCTION VALUES AND VARIABLE VALUES AT   */
/*                   EACH ITERATION AS WELL AS THE SEPARATION */
/*                   OF CONSTRAINTS INTO NONBINDING AND       */
/*                   BINDING AND VARIABLES INTO BASIC,        */
/*                   SUPERBASIC AND NONBASIC.                 */
/*                - 4 - AT EACH ITERATION THE REDUCED GRADIENT, */
/*                   THE SEARCH DIRECTION AND THE TANGENT     */
/*                   VECTOR ARE PRINTED.                      */
/*                - 5 - PROVIDES DETAILS OF THE BASIS INVERSION */
/*                   PROCESS INCLUDING THE INITIAL BASIS AND  */
/*                   ITS INVERSE.  ALSO DISPLAYS THE VARIABLE */
/*                   VALUES AND CONSTRAINT ERRORS FOR EACH    */
/*                   NEWTON ITERATION.                        */
/*                - 6 - THIS IS THE MAXIMUM LEVEL OF PRINT     */
/*                   AVAILABLE AND INCLUDES ALL OF THE ABOVE  */
/*                   ALONG WITH DETAILED PROGRESS OF THE      */
/*                   BASIS CONSTRUCTION PHASE, INCLUDING      */
/*                   THE BASIS INVERSE AT EACH PIVOT.         */

        grgsetparameter("ipr",(long) 0,(double)0.0);

/*   2  ipn4  -- 0     - IF  ipn# IS GREATER THAN ZERO THEN ipr  */
/*      ipn5    0        WILL BE SET TO # AFTER  ipn# ITERATIONS */
/*      ipn6    0                                              */

/*   grgsetparameter("ipn4",(long)-1,(double)0.0);              */

/*   3  iper  -- 0     - IF iper IS GREATER THAN ZERO THEN      */
```

```
/*                  FOR EVERY IPER-TH ITERATION, PRINT      */
/*                    USING THE CURRENT VALUE OF {ipr}   */
/*                    OTHERWISE USE ipr=1 .              */

/*     ioterm --    Enable/disable iteration log to screen      */
/*     (default) 0  No iteration log to screen                */
/*               1  Write iteration log to screen             */

/*   grgsetparameter("ioterm",(long)1,(double)0.0);                  */

/* METHODS  */
/*   1  iquad--- 0  - METHOD FOR INITIAL ESTIMATES OF BASIC       */
/*                    VARIABLES FOR EACH ONE DIMENSIONAL      */
/*                    SEARCH                                  */
/*              - 0 - TANGENT VECTOR AND LINEAR EXTRAPOLATION  */
/*                    WILL BE USED.                           */
/*              - 1 - QUADRATIC EXTRAPOLATION WILL BE USED.   */


/*   2  kderiv-- 0  - METHOD FOR OBTAINING PARTIAL DERIVATIVE     */
/*   (default)      - 0 - FORWARD DIFFERENCE APPROXIMATION       */
/*              - 1 - CENTRAL DIFFERENCE APPROXIMATION          */
/*              - 2 - USER SUPPLIED SUBROUTINE {parsh} IS USED */

                    grgsetparameter("kderiv",(long)1,(double)0.0);

/*   3  modcg -- 0  - {modcg} AND {maxrm} (SEE BELOW) CONTROL     */
/*                    USE OF A CONJUGATE GRADIENT ( CG )      */
/*                    METHOD.  IF THE NUMBER OF SUPERBASIC    */
/*                    VARIABLES EXCEEDS {maxrm}, THE CG       */
/*                    METHOD INDICATED BY {modcg} IS USED.    */
/*                    DEFAULT VALUE OF modcg=1 .  TO USE A    */
/*                    CG METHOD AT EACH ITERATION, SET        */
/*                    maxrm=0.                                */
/*              - 1 - USES FLETCHER-REEVES FORMULA.           */
/*              - 2 - USES POLAK-RIBIERE FORMULA.             */
/*              - 3 - USES PERRY'S FORMULA.                   */
/*              - 4 - USES 1 STEP VERSION OF DFP.             */
/*              - 5 - USES 1 STEP VERSION OF BFS.             */



/*   4  maxrm -- maxr - MAXIMUN NUMBER OF ROWS FOR HESSIAN APPROX  */
/*                    FOR THE BFGS ALGORITHM. IF THE NUMBER OF   */
/*                    SUPERBASICS EXCEEDS maxrm THEN A CONJUGATE */
/*                    GRADIENT ALGORITHM IS USED.  TO FORCE A CG */
/*                    METHOD TO BE ALWAYS USED SET maxrm = 0.    */



/*   5  maxim -- 0  - OBJECTIVE FUNCTION WILL BE MINIMIZED(default)*/
/*              - 1 - OBJECTIVE WILL BE MAXIMIZED               */

                    grgsetparameter("maxim",(long)1,(double)0.0);
```

```
/* ******************************************************************** */
/*     ++++++++++++++++++++++++++++++++++++                              */
/*     + OUTPUT VARIABLES AND PARAMETERS +                              */
/*     ++++++++++++++++++++++++++++++++++++                              */
/*   g     -- DOUBLE ARRAY WITH MAX SUBSCRIPT EQUAL TO {nrows}.  */
/*           g[i] CONTAINS FINAL VALUE OF FUNCTIONS IN GCOMP.   */
/*   xx    -- DOUBLE ARRAY WITH MAX SUBSCRIPT EQUAL TO {nvars}.  */
/*           xx[i] CONTAINS FINAL VALUE OF VARIABLE i.          */
/*   inbind -- INTEGER ARRAY WITH MAX SUBSCRIPT {nrows}. POSITIONS */
/*           1 TO nbind OF inbind CONTAIN THE INDICES OF        */
/*           THOSE FUNCTIONS (COMPONENTS OF THE g VECTOR) WHICH  */
/*           ARE AT THEIR LOWER OR UPPER BOUNDS AT TERMINATION.  */
/*   rmult --  DOUBLE ARRAY WITH MAX SUBSCRIPT {nrows}. POSITIONS  */
/*           1 TO nbind OF rmult CONTAIN LAGRANGE MULTIPLIERS    */
/*           OF THE BINDING CONSTRAINTS, CORRESPONDING TO THE    */
/*           INDICES IN INBIND.  */
/*   nonbas -- INTEGER ARRAY WITH MAX SUBSCRIPT {nvars}. POSITIONS */
/*           1 TO {nnonb} CONTAIN THE INDICES OF THOSE          */
/*           COMPONENTS OF xx WHICH ARE NOT BASIC (I.E. EITHER   */
/*           SUPERBASIC OR NONBASIC) AT TERMINATION. THE        */
/*           REMAINING POSITIONS CONTAIN NO USEFUL INFORMATION.  */
/*   redgr  -- DOUBLE ARRAY OF SAME SIZE AS nonbas. POSITIONS FROM */
/*           1 TO {nnonb} CONTAIN THE REDUCED GRADIENTS OF THE   */
/*           VARIABLES WHOSE INDICES ARE IN CORRESPONDING       */
/*           POSITIONS IN nonbas.                              */
/*   nbind  -- INTEGER SCALAR. NUMBER OF BINDING CONSTRAINTS. SEE  */
/*           DESCRIPTION OF inbind AND rmult ABOVE.            */
/*   nnonb  -- INTEGER. SEE nonbas AND redgr EXPLNANATION ABOVE    */
/*              ++++++++++++++++++++                              */
/*               TERMINATION CODES                               */
/*              ++++++++++++++++++++                              */
/*  The value retuned as the result of function grgsub indicates    */
/*  the reason for grg2's termination                              */
/*                                                                 */
/*  The termination codes for this version of GRGC are specified by  */
/*  the enum constant 'grg2_termination_codes'. The symbolic names   */
/*  for these codes are given below. Users can access these names    */
/*  by including the header file 'grgcodes.h'. A call to the        */
/*  function 'grg2_get_terminationmsg()' will return a pointer to    */
/*  a string containing the termination message printed in the       */
/*  FINAL RESULTS section.                                          */
/*                                                                 */
/*  NOTE:  The current numeric values of these codes are given below */
/*       for informational purposes. Use of these values in hard   */
/*       code is not advised as these values might change if more  */
/*       detailed diagnostics are added.                           */
/*                                                                 */
/*  Termination Code Name          Termination Condition          */
/*  ---------------------     ---------------------------------  */
/* _GRG2_INSFMEM   (-2)       Could not allocate sufficient memory */
/*                        to solve problem                        */
/* _GRG2_BADINPUT  (-1)       Errors in input options or problem   */
/*                        dimensions                              */
/* _GRG2_KTC       (1)        Kuhn-Tucker Conditions Satisified    */
/*                                                                 */
```

```
/* _GRG2_FRACTCHG  (2)        Fractional Change in Obj Function    */
/*                    <epstop> for <nstop> consecutive itns */
/*                                                    */
/* _GRG2_ALLREMEDIES  (3)      All Remedies Have Failed to Find     */
/*                    a better point                  */
/* _GRG2_ITERATIONS   (4)      Limit on One-Dimensional  Searches   */
/*                    <limser> exceeded                 */
/* _GRG2_UNBOUNDED    (5)      Solution Unbounded                 */
/*                                                    */
/*  The termination codes below indicate that GRG2 terminated during */
/*  Phase I and hence was unable to find a feasible point.  These    */
/*  codes indicate that the problem was infeasible as well as the    */
/*  reason that Phase I terminated.                          */
/*                                                    */
/* _GRG2_INFEASIBLE_KTC     (6)                          */
/*                                                    */
/* _GRG2_INFEASIBLE_FRACTCHG (7)                          */
/*                                                    */
/* _GRG2_INFEASIBLE_ALLREMEDIES (8)                          */
/*                                                    */
/* _GRG2_INFEASIBLE_ITERATIONS  (9)                          */
/*                                                    */
/* ************************************************************** */


  info = grgsub(nvars,nrows,nobj,title,inprint,outprint,var,con,xlb,xub,
            glb,gub,g,xx,rmult,redgr,inbind,nonbas,
            &nnonb,&nbind,ioout);
  printf("\n\n... return from grgsub ....");

  fprintf(ioout,
    "\n\n\n grgsub returned a code of %2d indicating:\n %s",info,
       grg2_get_terminationmsg() );

  fprintf(ioout,"\n\n Final Objective Value = %12.6g",g[nobj]);
  fprintf(ioout,"\n\n..... Final Variable Values are .....\n\n");
  for(i=1;i<=nvars;++i) {
     fprintf(ioout," x[%4d] = %12.6g",i,xx[i]);
     if(i%3==0) fprintf(ioout,"\n");
  }
}
```

awpcc.c

```
/*****************************************************************/
/*                          SUBROUTINE GCOMP                     */
/* THIS PROGRAM DOES THERMODYNAMIC CALCULATIONS OF THE           */
/* AMMONIA-BASED COMBINED POWER/COOLING CYCLE TO GET THE         */
/* OBJECTIVE FUNCTION, WHICH IS THERMAL EFFICIENCY, AND          */
/* CONSTRAINTS.                                                  */
/*****************************************************************/


#include <stdio.h>
#include <math.h>
#include "awpcc.h"
#include "property.c"


void boiler(double p_in, double x_in, double t_boiler, double t_cond,
                        double h_boiler[], double s_boiler[], double v_boiler[])
{

// ammonia mass fraction in boiler and condenser, x: liquid, y: vapor

 double  xs;
 double amv_boiler, aml_boiler, amv_cond, aml_cond, amv_rcond, aml_rcond;
 double h_re, s_re, v_re;
//double reflux, reboiler;

// incoming mixture property
 t_boilermin=bubble(p_in, x_in);
 if(t_boiler <= t_boilermin) return;
 amm_fraction(t_boiler, p_in, &x_boiler, &y_boiler);

 xs = x_in;
 amv_boiler = (xs - x_boiler)/(y_boiler - x_boiler);
 aml_boiler = (y_boiler - xs)/(y_boiler - x_boiler);


 t_condmin = bubble(p_in, y_boiler);
 if (t_cond <= t_condmin) return;
 amm_fraction(t_cond, p_in, &x_cond, &y_cond);

 amv_cond = (y_boiler - x_cond)/(y_cond - x_cond);
 aml_cond = (y_cond - y_boiler)/(y_cond - x_cond);

 amv_rcond = (x_cond - x_boiler)/(y_boiler - x_boiler);
 aml_rcond = (y_boiler - x_cond)/(y_boiler - x_boiler);

 mix_l(t_boiler, p_in, x_boiler, &h_re, &s_re, &v_re);
 h_boiler[2] = h_re;
 s_boiler[2] = s_re;
 v_boiler[2] = v_re;
```

```
  mix_v(t_boiler, p_in, y_boiler, &h_re, &s_re, &v_re);
  h_boiler[3] = h_re;
  s_boiler[3] = s_re;
  v_boiler[3] = v_re;
  mix_l(t_cond, p_in, x_cond, &h_re, &s_re, &v_re);
  h_boiler[4] = h_re;
  s_boiler[4] = s_re;
  v_boiler[4] = v_re;
  mix_v(t_cond, p_in, y_cond, &h_re, &s_re, &v_re);
  h_boiler[5] = h_re;
  s_boiler[5] = s_re;
  v_boiler[5] = v_re;

// reflux   = aml_cond / amv_cond;
// reboiler = aml_boiler /  amv_boiler;

  f4 = amv_boiler / (1 - aml_cond * amv_rcond);
  f5 = f4 * aml_cond;
  fa = f4 * amv_cond;
  fw = aml_boiler + f5 * aml_rcond;

  return;
}


void gcomp(double g[], double z[])
{
  double ms = 1. ,x_com;
  double t_ab, t_boiler, t_cond, t_super, p_high, p_low;
  double h[15], s[15], v[15], t[15], p[15], x[15], f[15];
  double h_boiler[6], s_boiler[6], v_boiler[6];
  double q_absorber, q_boiler, q_cond, q_super, q_cooler;
  double wt, eff, pump;
  double x_turbine, y_turbine;
  double amv_turbine, aml_turbine;
  double molehs, phs;
  double ths1, hhs1, shs1, vhs1;
  double ths2, hhs2, shs2, vhs2;
  double ths3, hhs3, shs3, vhs3;
  double thsm, hhsm, shsm, vhsm;
  double t_pin, h_pin, s_pin, v_pin, q_pin;
  double t0, p0, h0, s0, v0;
  double exhs, effex;
  double deltat;
  int j;

  t_ab    = z[1];
  t_boiler = z[2];
  t_cond  = z[3];
  t_super  = z[4];
  p_high  = z[5];
  x_com   = z[6];
  ths1    = z[7];
  ths2    = z[8];

  p_low = sat_p(t_ab, x_com);
```

```
//point 1
        t[1] = t_ab;
        p[1] = p_low;
        x[1] = x_com;
        f[1] = 1.;
        mix_l(t[1], p[1], x[1], &h[1], &s[1], &v[1]);

//point 2
        p[2] = p_high;
        x[2] = x_com;
        s[2] = s[1];
        f[2] = 1.;
        property_s_t(p[2], s[2], x[2], &h[2], &t[2], &v[2]);


 boiler(p_high,x_com,t_boiler,t_cond,h_boiler,s_boiler,v_boiler);
 if(t_boiler <= t_boilermin || t_cond <= t_condmin)
 {
   for(j = 1; j <= 13; j++)
          g[j] = 1.0e+30;
       return;
 }


//point 10
        t[10] = t_boiler;
        p[10] = p_high;
        x[10] = x_boiler;
        f[10] = fw;
        h[10] = h_boiler[2];
        s[10] = s_boiler[2];
        v[10] = v_boiler[2];

//point 11
        t[11] = t[2] + 5.;
        p[11] = p_high;
        x[11] = x_boiler;
        f[11] = fw;
        property(t[11], p[11], x[11], &h[11], &s[11], &v[11]);

//point 12
        p[12] = p_low;
        x[12] = x_boiler;
        f[12] = fw;
        h[12] = h[11];
        property_h_t(p[12], h[12], x[12], &t[12], &s[12], &v[12]);

//point 4
        t[4] = t_boiler;
        p[4] = p_high;
        f[4] = f4;
        h[4] = h_boiler[3];
        s[4] = s_boiler[3];
        v[4] = v_boiler[3];
        x[4] = y_boiler;
```

```
//point 5
        t[5] = t_cond;
        p[5] = p_high;
        f[5] = f5;
        h[5] = h_boiler[4];
        s[5] = s_boiler[4];
        v[5] = v_boiler[4];
        x[5] = x_cond;

//point 6
        t[6] = t_cond;
        p[6] = p_high;
        x[6] = y_cond;
        f[6] = fa;
        h[6] = h_boiler[5];
        s[6] = s_boiler[5];
        v[6] = v_boiler[5];

  q_cond   = ms * (f[6] * h[6] + f[5] * h[5] - f[4] * h[4]);

//point 13
        t[13] = t_cond - 5.;
        p[13] = p_high;
        x[13] = x_com;
        property(t[13], p[13], x[13], &h[13], &s[13], &v[13]);
        f[13] = -q_cond / (ms * (h[13] - h[2]));

//point 14
        p[14] = p_high;
        x[14] = x_com;
        f[14] = 1. - f[13];
        h[14] = h[2] + (h[10] - h[11]) * fw / f[14];
        property_h_t(p[14], h[14], x[14], &t[14], &s[14], &v[14]);

//point 3
        p[3] = p_high;
        x[3] = x_com;
        f[3] = 1.;
        h[3] = (f[13] * h[13] + f[14] * h[14]) / f[3];
        property_h_t(p[3], h[3], x[3], &t[3], &s[3], &v[3]);

  q_boiler = ms * (f[4] * h[4] + f[10] * h[10] - f[3] * h[3] - f[5] * h[5]);

//point 7
        t[7] = t_super;
        p[7] = p_high;
        x[7] = y_cond;
        f[7] = fa;
        mix_v(t[7], p[7], x[7], &h[7], &s[7], &v[7]);

//point 8
        p[8] = p_low;
        s[8] = s[7];
        x[8] = y_cond;
        f[8] = fa;
```

```
        property_s_t(p[8], s[8], x[8], &h[8], &t[8], &v[8]);
        amm_fraction(t[8], p_low, &x_turbine, &y_turbine);
        amv_turbine = (y_cond - x_turbine)/(y_turbine - x_turbine);
        aml_turbine = (y_turbine - y_cond)/(y_turbine - x_turbine);

//point 9
        p[9] = p_low;
        x[9] = y_cond;
        f[9] = fa;
        if(t[8] < t_ab)
          {
                t[9] = t_ab;
                property(t[9], p[9], x[9], &h[9], &s[9], &v[9]);
                q_cooler = ms * f[8] * (h[9] - h[8]);
          }
        else
          {
                q_cooler = 0.;
                t[9] = t[8];
                h[9] = h[8];
                s[9] = s[8];
                v[9] = v[8];
          }

  q_super = ms * f[6] * (h[7] - h[6]);
  q_absorber = ms * (f[1] * h[1] - f[12] * h[12] - f[9] * h[9]);
  wt = ms * f[7] * (h[7] - h[8]);
  pump = ms * f[1] * (h[2] - h[1]);
  eff = (wt-pump+q_cooler) / (q_super + q_boiler) * 100;

  phs = h2o_sat_p(ths1);
  if(phs < 1.013) phs = 1.013;
  steam_l(ths1, phs, &hhs1, &shs1, &vhs1);
  steam_l(ths2, phs, &hhs2, &shs2, &vhs2);

  molehs = (q_super + q_boiler)/(hhs1 - hhs2);
  hhs3 = hhs1 - q_super/molehs;
  steam_l_ph(phs, hhs3, &ths3, &shs3, &vhs3);

  t_pin = t_boilermin;
  property(t_pin, p_high, x_com, &h_pin, &s_pin, &v_pin);
  q_pin = ms * (h_pin - h[3]);

  hhsm = hhs2 + q_pin/molehs;
  steam_l_ph(phs, hhsm, &thsm, &shsm, &vhsm);
  deltat = thsm-t_pin;

  t0 = 330.;
  p0 = 1.013;
  steam_l(t0, p0, &h0, &s0, &v0);
  exhs=molehs*((hhs1-h0)-t0*(shs1-s0));

  effex = (wt-pump+q_cooler) / exhs * 100.;

  g[1] = t_cond - t_boiler;
  g[2] = t_cond - t_super;
```

```
 g[3] = t_boilermin - t_boiler;
 g[4] = fa * 100.;
 g[5] = t_condmin - t_cond;
 g[6] = t[10] - t[14];
 g[7] = amv_turbine * 100.;
 g[8] = f[13] * 100.;
 g[9] = ths1 - t[7];
 g[10]= ths2 - t[3];
 g[11]= ths3 - t[4];
 g[12]= deltat;
 g[13]= eff;
}
```

awpcc.h

```
double x_boiler, y_boiler, x_cond, y_cond;
double t_boilermin, t_condmin;
double f4, f5, fw, fa;
```

property.c

```
/*************************************************************************/
/*                          PROPERTY PROGRAM                             */
/* THIS PROGRAM CALCULATES THE PROPERTY OF THE AMMONIA/WATER             */
/* MIXTURES                                                              */
/*************************************************************************/

#include <math.h>
#include "property.h"


void steam_l(double t, double p, double *h_h2ol, double *s_h2ol, double *v_h2ol)
{
 double h, s, v;

 Tr = t/Tb;
 Pr = p/Pb;
 h = -h2ohl_ro + h2o_b1*h2oT_ro + h2o_b2/2*(h2oT_ro*h2oT_ro + Tr*Tr)
         + h2o_b3/3*(h2oT_ro*h2oT_ro*h2oT_ro - Tr*Tr*Tr) - h2o_b1*Tr
         - h2o_b2*Tr*Tr + (h2o_a4*Tr*Tr - h2o_a1)*(Pr - h2oP_ro)
         - h2o_a2/2*(Pr*Pr - h2oP_ro*h2oP_ro);
 *h_h2ol = -r*Tb*h;

 s = -h2osl_ro - h2o_b1*log(Tr/h2oT_ro) + h2o_b2*(h2oT_ro - Tr)
         + h2o_b3/2*(h2oT_ro*h2oT_ro - Tr*Tr) + (h2o_a3 + 2*h2o_a4*Tr)*(Pr - h2oP_ro);
 *s_h2ol = -r*s;

 v = h2o_a1 + h2o_a2*Pr + h2o_a3*Tr + h2o_a4*Tr*Tr;
 *v_h2ol = r*Tb/Pb*v;
}


void steam_v(double t, double p, double *h_h2ov, double *s_h2ov, double *v_h2ov)
{
 double h, s, v;

 Tr = t/Tb;
 Pr = p/Pb;
 h = -h2ohg_ro + h2o_d1*h2oT_ro + h2o_d2/2*(h2oT_ro*h2oT_ro + Tr*Tr)
         + h2o_d3/3*(h2oT_ro*h2oT_ro*h2oT_ro - Tr*Tr*Tr) - h2o_d1*Tr
         - h2o_d2*Tr*Tr - h2o_c1*(Pr - h2oP_ro)
         + h2o_c2*4*(h2oP_ro*pow(h2oT_ro,-3) - Pr*pow(Tr,-3))
         + h2o_c3*12*(h2oP_ro*pow(h2oT_ro,-11) - Pr*pow(Tr,-11))
         + h2o_c4*4*(pow(h2oP_ro,3)*pow(h2oT_ro,-11) - pow(Pr,3)*pow(Tr,-11));
 *h_h2ov = -r*Tb*h;

 s = -h2osg_ro - h2o_d1*log(Tr/h2oT_ro) + h2o_d2*(h2oT_ro - Tr)
         + h2o_d3/2*(h2oT_ro*h2oT_ro - Tr*Tr) + log(Pr/h2oP_ro)
         + 3*h2o_c2*(h2oP_ro*pow(h2oT_ro,-4) - Pr*pow(Tr,-4))
         + 11*h2o_c3*(h2oP_ro*pow(h2oT_ro,-12) - Pr*pow(Tr,-12))
         + 11/3*h2o_c4*(pow(h2oP_ro,3)*pow(h2oT_ro,-12) - pow(Pr,3)*pow(Tr,-12));
```

```
 *s_h2ov = -r*s;

 v = Tr/Pr + h2o_c1 + h2o_c2*pow(Tr,-3) + pow(Tr,-11)*(h2o_c3 + h2o_c4*Pr*Pr);
 *v_h2ov = r*Tb/Pb*v;
}


void ammonia_l(double t, double p, double *h_h3nl, double *s_h3nl, double *v_h3nl)
{
 double h, s, v;

 Tr = t/Tb;
 Pr = p/Pb;
 h = -h3nhl_ro + h3n_b1*h3nT_ro + h3n_b2/2*(h3nT_ro*h3nT_ro + Tr*Tr)
         + h3n_b3/3*(h3nT_ro*h3nT_ro*h3nT_ro - Tr*Tr*Tr) - h3n_b1*Tr
         - h3n_b2*Tr*Tr + (h3n_a4*Tr*Tr - h3n_a1)*(Pr - h3nP_ro)
         - h3n_a2/2*(Pr*Pr - h3nP_ro*h3nP_ro);
 *h_h3nl = -r*Tb*h;

 s = -h3nsl_ro - h3n_b1*log(Tr/h3nT_ro) + h3n_b2*(h3nT_ro - Tr)
         + h3n_b3/2*(h3nT_ro*h3nT_ro - Tr*Tr) + (h3n_a3 + 2*h3n_a4*Tr)*(Pr - h3nP_ro);
 *s_h3nl = -r*s;

 v = h3n_a1 + h3n_a2*Pr + h3n_a3*Tr + h3n_a4*Tr*Tr;
 *v_h3nl = r*Tb/Pb*v;
}


void ammonia_v(double t, double p, double *h_h3nv, double *s_h3nv, double *v_h3nv)
{
 double h, s, v;

 Tr = t/Tb;
 Pr = p/Pb;
 h = -h3nhg_ro + h3n_d1*h3nT_ro + h3n_d2/2*(h3nT_ro*h3nT_ro + Tr*Tr)
         + h3n_d3/3*(h3nT_ro*h3nT_ro*h3nT_ro - Tr*Tr*Tr) - h3n_d1*Tr
         - h3n_d2*Tr*Tr - h3n_c1*(Pr - h3nP_ro)
         + h3n_c2*4*(h3nP_ro*pow(h3nT_ro,-3) - Pr*pow(Tr,-3))
         + h3n_c3*12*(h3nP_ro*pow(h3nT_ro,-11) - Pr*pow(Tr,-11))
         + h3n_c4*4*(pow(h3nP_ro,3)*pow(h3nT_ro,-11) - pow(Pr,3)*pow(Tr,-11));
 *h_h3nv = -r*Tb*h;

 s = -h3nsg_ro - h3n_d1*log(Tr/h3nT_ro) + h3n_d2*(h3nT_ro - Tr)
         + h3n_d3/2*(h3nT_ro*h3nT_ro - Tr*Tr) + log(Pr/h3nP_ro)
         + 3*h3n_c2*(h3nP_ro*pow(h3nT_ro,-4) - Pr*pow(Tr,-4))
         + 11*h3n_c3*(h3nP_ro*pow(h3nT_ro,-12) - Pr*pow(Tr,-12))
         + 11/3*h3n_c4*(pow(h3nP_ro,3)*pow(h3nT_ro,-12) - pow(Pr,3)*pow(Tr,-12));
 *s_h3nv = -r*s;

 v = Tr/Pr + h3n_c1 + h3n_c2*pow(Tr,-3) + pow(Tr,-11)*(h3n_c3 + h3n_c4*Pr*Pr);
 *v_h3nv = r*Tb/Pb*v;
}


void excess(double t, double p, double x, double *he, double *se, double *ve)
{
```

```
 double f1, f2, f3;

 x = (x/wm2)/((1 - x)/wm1 + x/wm2);
 Tr = t/Tb;
 Pr = p/Pb;
 f1 = e1 + e2*Pr + 2*e5/Tr +3*e6/(Tr*Tr);
 f2 = e7 + e8*Pr + 2*e11/Tr +3*e12/(Tr*Tr);
 f3 = e13 + e14*Pr + 2*e15/Tr +3*e16/(Tr*Tr);

 *he = r*Tb*(f1 + f2*(2*x - 1) + f3*(2*x - 1)*(2*x - 1))*(1 - x)*x;

 f1 = e3 + e4*Pr - e5/(Tr*Tr) - 2*e6/(Tr*Tr*Tr);
 f2 = e9 + e10*Pr - e11/(Tr*Tr) - 2*e12/(Tr*Tr*Tr);
 f3 = -e15/(Tr*Tr) - 2*e16/(Tr*Tr*Tr);

 *se = -r*(f1 + f2*(2*x - 1) + f3*(2*x - 1)*(2*x - 1))*(1 - x)*x;
 *ve = r*Tb/Pb*(e2 + e4*Tr +(e8 + e10*Tr)*(2*x - 1)
          + e14*(2*x - 1)*(2*x - 1))*(1 - x)*x;
}


double bubble(double p, double xf)
{
 double tcw, tcb, pcb, pcw;
 double pz;

 tcw = (tc1 - 273.15)*1.8 + 32;
 pcw = pc1 * 14.50377;
 tcb = tcw - (a1*xf + a2*pow(xf,2) + a3*pow(xf,3) + a4*pow(xf,4));
 pcb = pcw*exp(b1*xf + b2*pow(xf,2) + b3*pow(xf,3) + b4*pow(xf,4)
          + b5*pow(xf,5) + b6*pow(xf,6) + b7*pow(xf,7) + b8*pow(xf,8));
 pz = log(pcb/(p*14.50377));
 tb1 = (c1 + (c11*xf + c12*pow(xf,2) + c13*pow(xf,3) + c14*pow(xf,4)
          + c15*pow(xf,5) + c16*pow(xf,6) + c17*pow(xf,7) + c18*pow(xf,8)
          + c19*pow(xf,9) + c110*pow(xf,10)))*pow(pz,1);
 tb2 = (c2 + (c21*xf + c22*pow(xf,2) + c23*pow(xf,3) + c24*pow(xf,4)
          + c25*pow(xf,5) + c26*pow(xf,6) + c27*pow(xf,7) + c28*pow(xf,8)
          + c29*pow(xf,9) + c210*pow(xf,10)))*pow(pz,2);
 tb3 = (c3 + (c31*xf + c32*pow(xf,2) + c33*pow(xf,3) + c34*pow(xf,4)
          + c35*pow(xf,5) + c36*pow(xf,6) + c37*pow(xf,7) + c38*pow(xf,8)
          + c39*pow(xf,9) + c310*pow(xf,10)))*pow(pz,3);
 tb4 = (c4 + (c41*xf + c42*pow(xf,2) + c43*pow(xf,3) + c44*pow(xf,4)
          + c45*pow(xf,5) + c46*pow(xf,6) + c47*pow(xf,7) + c48*pow(xf,8)
          + c49*pow(xf,9) + c410*pow(xf,10)))*pow(pz,4);
 tb5 = (c5 + (c51*xf + c52*pow(xf,2) + c53*pow(xf,3) + c54*pow(xf,4)
          + c55*pow(xf,5) + c56*pow(xf,6) + c57*pow(xf,7) + c58*pow(xf,8)
          + c59*pow(xf,9) + c510*pow(xf,10)))*pow(pz,5);
 tb6 = (c6 + (c61*xf + c62*pow(xf,2) + c63*pow(xf,3) + c64*pow(xf,4)
          + c65*pow(xf,5) + c66*pow(xf,6) + c67*pow(xf,7) + c68*pow(xf,8)
          + c69*pow(xf,9) + c610*pow(xf,10)))*pow(pz,6);
 tb7 = (c7 + (c71*xf + c72*pow(xf,2) + c73*pow(xf,3) + c74*pow(xf,4)
          + c75*pow(xf,5) + c76*pow(xf,6) + c77*pow(xf,7) + c78*pow(xf,8)
          + c79*pow(xf,9) + c710*pow(xf,10)))*pow(pz,7);
 tb = tcb - (tb1 + tb2 + tb3 + tb4 + tb5 + tb6 + tb7);

 return tb = (tb - 32)/1.8 + 273.15;
```

```
}


double dew(double p, double xg)
{
 double tcw, pcw, px, aa, s1, s2;
 double pc, tc;

 tcw = (tc1 - 273.15)*1.8 + 32;
 tc = tcw - (a1*xg + a2*pow(xg,2) + a3*pow(xg,3) + a4*pow(xg,4));
 pcw = pc1*14.50377;
 s1 = b1*xg + b2*pow(xg,2) + b3*pow(xg,3) + b4*pow(xg,4);
 s2 = b5*pow(xg,5) + b6*pow(xg,6) + b7*pow(xg,7) + b8*pow(xg,8);
 pc = pcw*exp(s1 + s2);
 px = log(pc/(p*14.50377));
 aa = log(1.0001 - xg);
 td1 = (a11 + (a11*aa + a12*pow(aa,2) + a13*pow(aa,3) + a14*pow(aa,4)))*pow(px,1);
 td2 = (a12 + (a21*aa + a22*pow(aa,2) + a23*pow(aa,3) + a24*pow(aa,4)))*pow(px,2);
 td3 = (a13 + (a31*aa + a32*pow(aa,2) + a33*pow(aa,3) + a34*pow(aa,4)))*pow(px,3);
 td4 = (a14 + (a41*aa + a42*pow(aa,2) + a43*pow(aa,3) + a44*pow(aa,4)))*pow(px,4);
 td5 = (a15 + (a51*aa + a52*pow(aa,2) + a53*pow(aa,3) + a54*pow(aa,4)))*pow(px,5);
 td6 = (a16 + (a61*aa + a62*pow(aa,2) + a63*pow(aa,3) + a64*pow(aa,4)))*pow(px,6);

 td = tc - (td1 + td2 + td3 + td4 + td5 + td6);

 return td = (td - 32)/1.8 + 273.15;
}


void amm_fraction(double t, double p, double *xf, double *xg)
{
 double err, xf1, xf2, xfold, xg1, xg2, xgold;

 xf1=0.0;  xf2=1.0;
 *xf=0.0;
 do
        {
        xfold=*xf;
        *xf=(xf1+xf2)/2.;
        tb=bubble(p,*xf);
        if(tb<t) xf2=*xf;
        else xf1=*xf;
        err=fabs(*xf-xfold);
        }while(err>1.0e-20);

 xg1=0.0;  xg2=1.0;
 *xg=0.0;
 do
        {
        xgold=*xg;
        *xg=(xg1+xg2)/2.;
        td=dew(p,*xg);
        if(td<t) xg2=*xg;
        else xg1=*xg;
        err=fabs(*xg-xgold);
        }while(err>1.0e-20);
```

```
}


void mix_l(double t, double p, double x, double *hmix, double *smix, double *vmix)
{
 double frmolw, frmola;
 double wm;

 steam_l(t,p,&h_h2ol,&s_h2ol,&v_h2ol);
 ammonia_l(t, p,&h_h3nl,&s_h3nl,&v_h3nl);
 excess(t, p, x, &he, &se, &ve);
 frmolw = ((1 - x)/wm1)/((1 - x)/wm1 + x/wm2);
 frmola = 1 - frmolw;
 wm = frmolw * wm1 + frmola * wm2;
 *hmix = h_h2ol*frmolw + h_h3nl*frmola + he;
 if(frmola == 0. || frmolw == 0.)
   *smix = s_h2ol*frmolw + s_h3nl*frmola;
 else
   *smix = s_h2ol*frmolw + s_h3nl*frmola + se - r*(frmola*log(frmola) + frmolw*log(frmolw));
 *vmix = v_h2ol*frmolw + v_h3nl*frmola + ve;

 *hmix = *hmix/wm;
 *smix = *smix/wm;
 *vmix = *vmix/wm;
}


void mix_v(double t, double p, double x, double *hmix, double *smix, double *vmix)
{
 double frmolw, frmola;
 double wm;

 steam_v(t,p,&h_h2ov,&s_h2ov,&v_h2ov);
 ammonia_v(t, p,&h_h3nv,&s_h3nv,&v_h3nv);
 frmolw = ((1 - x)/wm1)/((1 - x)/wm1 + x/wm2);
 frmola = 1 - frmolw;
 wm = frmolw * wm1 + frmola * wm2;

 *hmix = h_h2ov*frmolw + h_h3nv*frmola;
 if(frmola == 0. || frmolw == 0.)
   *smix = s_h2ov*frmolw + s_h3nv*frmola;
 else
   *smix = s_h2ov*frmolw + s_h3nv*frmola - r*(frmola*log(frmola) + frmolw*log(frmolw));
 *vmix = v_h2ov*frmolw + v_h3nv*frmola;

 *hmix = *hmix/wm;
 *smix = *smix/wm;
 *vmix = *vmix/wm;
}


void mix_equ(double t, double p, double x, double *hmix, double *smix, double *vmix)
{
 double amv, aml;
 double h2omfv, h2omfl, h3nmfv, h3nmfl;
 double xs, xf, xg, xfw, xgw, wml, wmv;
```

```
double hmixl, hmixv, smixl, smixv, vmixl, vmixv;

steam_l(t,p,&h_h2ol,&s_h2ol,&v_h2ol);
ammonia_l(t, p,&h_h3nl,&s_h3nl,&v_h3nl);
steam_v(t,p,&h_h2ov,&s_h2ov,&v_h2ov);
ammonia_v(t, p,&h_h3nv,&s_h3nv,&v_h3nv);

amm_fraction(t, p, &xf, &xg);
xfw = 1 - xf;
xgw = 1 - xg;
xs = x;
amv = (xs - xf)/(xg - xf);
aml = (xg - xs)/(xg - xf);
h2omfv = (amv*xgw/wm1)/(amv*xgw/wm1 + amv*xg/wm2);
h2omfl = (aml*xfw/wm1)/(aml*xfw/wm1 + aml*xf/wm2);
h3nmfv = 1 - h2omfv;
h3nmfl = 1 - h2omfl;
wml = h2omfl * wm1 + h3nmfl * wm2;
wmv = h2omfv * wm1 + h3nmfv * wm2;

excess(t, p, xf, &he, &se, &ve);

hmixl = h_h2ol*h2omfl + h_h3nl*h3nmfl + he;
if(h3nmfl == 0. || h2omfl == 0.)
  smixl = s_h2ol*h2omfl + s_h3nl*h3nmfl;
else
  smixl = s_h2ol*h2omfl + s_h3nl*h3nmfl + se - r*(h3nmfl*log(h3nmfl) + h2omfl*log(h2omfl));
vmixl = v_h2ol*h2omfl + v_h3nl*h3nmfl + ve;
hmixl = hmixl/wml;
smixl = smixl/wml;
vmixl = vmixl/wml;

hmixv = h_h2ov*h2omfv + h_h3nv*h3nmfv;
if(h3nmfv == 0. || h2omfv == 0.)
  smixv = s_h2ov*h2omfv + s_h3nv*h3nmfv;
else
  smixv = s_h2ov*h2omfv + s_h3nv*h3nmfv - r*(h3nmfv*log(h3nmfv) + h2omfv*log(h2omfv));
vmixv = v_h2ov*h2omfv + v_h3nv*h3nmfv;
hmixv = hmixv/wmv;
smixv = smixv/wmv;
vmixv = vmixv/wmv;

*hmix = amv * hmixv + aml * hmixl;
*smix = amv * smixv + aml * smixl;
*vmix = amv * vmixv + aml * vmixl;
}


void property(double t, double p, double x, double *hmix, double *smix, double *vmix)
{
td = dew(p, x);
tb = bubble(p, x);

if (t <= tb)       mix_l(t, p, x, hmix, smix, vmix);
else if (t >= td)  mix_v(t, p, x, hmix, smix, vmix);
else               mix_equ(t, p, x, hmix, smix, vmix);
```

```
}


void property_s_t(double p, double s_st, double x, double *hmixs, double *tts, double *vmixs)
{
 double tt1, tt2, ttsold, err, *smixs, hercules=0;
 double hb, sb, vb;
 double hd, sd, vd;
 double xf, xg, xmixs;
 double aml, amv;
 double address1=0, address2=0, address3=0, address4=0, address5=0, address6=0;
 double *hmixf, *smixf, *vmixf;
 double *hmixg, *smixg, *vmixg;


 *tts=0.;
 *hmixs=0.;
 smixs=&hercules;
 *vmixs=0.;
 hmixf=&address1;
 smixf=&address2;
 vmixf=&address3;
 hmixg=&address4;
 smixg=&address5;
 vmixg=&address6;


 td = dew(p, x);
 tb = bubble(p, x);
 mix_l(tb, p, x, &hb, &sb, &vb);
 mix_v(td, p, x, &hd, &sd, &vd);


 if(s_st <= sb)
   {
    tt1=200.;
    tt2=tb;
    do{
       ttsold=*tts;
       *tts = (tt1 + tt2)/2.;
       mix_l(*tts, p, x, hmixs, smixs, vmixs);

       if (*smixs > s_st) tt2 = *tts;
              else        tt1 = *tts;
              err = fabs(*tts - ttsold);
         }while(err > 1.0e-20);
  }

 else if(s_st >= sd)
  {
    tt1=td;
    tt2=700.;
    do{
       ttsold=*tts;
       *tts = (tt1 + tt2)/2.;
       mix_v(*tts, p, x, hmixs, smixs, vmixs);
```

```
      if (*smixs > s_st) tt2 = *tts;
            else          tt1 = *tts;
            err = fabs(*tts - ttsold);
          }while(err > 1.0e-20);
   }

 else
  {
   tt1=tb;
   tt2=td;
   do{
      ttsold = *tts;
      *tts = (tt1 + tt2)/2.;

                  amm_fraction(*tts, p, &xf, &xg);
                  mix_l(*tts, p, xf, hmixf, smixf, vmixf);
                  mix_v(*tts, p, xg, hmixg, smixg, vmixg);
      amv = (s_st - *smixf) / (*smixg - *smixf);
                  aml = 1. - amv;
                  *hmixs = aml * (*hmixf) + amv * (*hmixg);
                  *vmixs = aml * (*vmixf) + amv * (*vmixg);
      xmixs = aml * xf    + amv * xg;

      if (xmixs > x)  tt1 = *tts;
      else          tt2 = *tts;
           err = fabs(*tts - ttsold);
         }while(err > 1.0e-20);
  }
}


void property_h_t(double p, double h_st, double x, double *tth, double *smixh, double *vmixh)
{
 double tt1, tt2, tthold, err, *hmixh, hercules=0;
 double hb, sb, vb;
 double hd, sd, vd;
 double xf, xg, xmixh;
 double aml, amv;
 double address1=0, address2=0, address3=0, address4=0, address5=0, address6=0;
 double *hmixf, *smixf, *vmixf;
 double *hmixg, *smixg, *vmixg;


 *tth=0.;
 hmixh=&hercules;
 *smixh=0.;
 *vmixh=0.;
 hmixf=&address1;
 smixf=&address2;
 vmixf=&address3;
 hmixg=&address4;
 smixg=&address5;
 vmixg=&address6;
```

```
td = dew(p, x);
tb = bubble(p, x);
mix_l(tb, p, x, &hb, &sb, &vb);
mix_v(td, p, x, &hd, &sd, &vd);


if(h_st <= hb)
  {
    tt1=200.;
    tt2=tb;
    do{
       tthold=*tth;
       *tth = (tt1 + tt2)/2.;
       mix_l(*tth, p, x, hmixh, smixh, vmixh);

      if (*hmixh > h_st) tt2 = *tth;
              else               tt1 = *tth;
              err = fabs(*tth - tthold);
         }while(err > 1.0e-20);
  }

else if(h_st >= hd)
  {
    tt1=td;
    tt2=700.;
    do{
       tthold=*tth;
       *tth = (tt1 + tt2)/2.;
       mix_v(*tth, p, x, hmixh, smixh, vmixh);

      if (*hmixh > h_st) tt2 = *tth;
              else               tt1 = *tth;
              err = fabs(*tth - tthold);
         }while(err > 1.0e-20);
  }

else
  {
    tt1=tb;
    tt2=td;
    do{
       tthold=*tth;
       *tth = (tt1 + tt2)/2.;

                    amm_fraction(*tth, p, &xf, &xg);
                    mix_l(*tth, p, xf, hmixf, smixf, vmixf);
                    mix_v(*tth, p, xg, hmixg, smixg, vmixg);
       amv = (h_st - *hmixf) / (*hmixg - *hmixf);
                    aml = 1. - amv;
                    *smixh = aml * (*smixf) + amv * (*smixg);
                    *vmixh = aml * (*vmixf) + amv * (*vmixg);
       xmixh = aml * xf    + amv * xg;

       if (xmixh > x)  tt1 = *tth;
       else            tt2 = *tth;
              err = fabs(*tth - tthold);
```

```
        }while(err > 1.0e-20);
    }
}


double sat_p(double t,double x)
{
 double p, p1, p2, pold, tt, err;

 p=0.;
 p1=0.;
 p2=10.;

 do
 {
  pold=p;
  p=(p1+p2)/2;
        tt=bubble(p,x);
        if(t>tt) p1=p;
        else     p2=p;
        err=fabs(p-pold);
        }while(err>1.0e-20);
 return p;
}


double h2o_sat_p(double t)
{
 double f[10] = {0.0, -7.691234564, -2.608023696e1, -1.681706546e2,
            6.423285504e1, -1.189646225e2, 4.16711732,
                                        2.09750676e1, 1.0e9, 6.0};
 double zt, ztt, pk, psk;
 int i;

 zt = t / 647.3;
 ztt = 1. - zt;
 pk = 0.;

 for(i = 5; i >= 0; i--) pk = pk * ztt + f[i];
 pk = pk / (zt * (1. + f[6] * ztt + f[7] * ztt * ztt));
 pk = pk - ztt / (f[8] * ztt * ztt + f[9]);
 psk = exp(pk) * 221.20;
 return psk;
}


void steam_1_ph(double p, double h, double *t_h2ol, double *s_h2ol, double *v_h2ol)
{
 double tt1=275, tt2=500, tt, ttold;
 double hh, ss, vv;
 double err;

 tt=hh=ss=vv=0;
 do
 {
  ttold=tt;
```

```
  tt=(tt1+tt2)/2;
  steam_l(tt, p, &hh, &ss, &vv);
  if(hh>h) tt2=tt;
  else     tt1=tt;
  err=fabs(tt-ttold);
  }while(err>1.0e-20);

 *t_h2ol=tt;
 *s_h2ol=ss;
 *v_h2ol=vv;
}
```

property.h

```
double h2o_a1 = 2.748796e-2, h2o_a2 = -1.016665e-5, h2o_a3 = -4.452025e-3, h2o_a4 = 8.389246e-4;
double h3n_a1 = 3.971423e-2, h3n_a2 = -1.790557e-5, h3n_a3 = -1.308905e-2, h3n_a4 = 3.752836e-3;

double h2o_b1 = 12.14557, h2o_b2 = -1.898065, h2o_b3 = 0.2911966;
double h3n_b1 = 16.34519, h3n_b2 = -6.508119, h3n_b3 = 1.448937;

double h2o_c1 = 2.136131e-2, h2o_c2 = -31.69291, h2o_c3 = -4.634611e+4, h2o_c4 = 0.0;
double h3n_c1 = -1.049377e-2, h3n_c2 = -8.288224, h3n_c3 = -6.647257e+2, h3n_c4 = -3.045352e+3;

double h2o_d1 = 4.019170, h2o_d2 = -5.175550e-2, h2o_d3 = 1.951939e-2;
double h3n_d1 = 3.673647, h3n_d2 = 9.989629e-2, h3n_d3 = 3.617622e-2;

double h2ohl_ro = 21.821141, h2ohg_ro = 60.965058;
double h3nhl_ro = 4.878573, h3nhg_ro = 26.468873;

double h2osl_ro = 5.733498, h2osg_ro = 13.453430;
double h3nsl_ro = 1.644773, h3nsg_ro = 8.339026;
double h2oT_ro = 5.0705, h2oP_ro = 3.0000;
double h3nT_ro = 3.2252, h3nP_ro = 2.0000;
double Tb = 100., Pb = 10.;

double e1 = -41.733398;
double e2 = 0.02414;
double e3 = 6.702285;
double e4 = -0.011475;
double e5 = 63.608967;
double e6 = -62.490768;
double e7 = 1.761064;
double e8 = 0.008626;
double e9 = 0.387983;
double e10 = -0.004772;
double e11 = -4.648107;
double e12 = 0.836376;
double e13 = -3.553627;
double e14 = 0.000904;
double e15 = 24.3616723;
double e16 = -20.736547;


double c1 = 153.634521459;
double c2 = -13.0305543892;
double c3 = -1.14845282991;
double c4 = .550358094447;
double c5 = -.0753450148427;
double c6 = .0048111666267;
double c7 = -.000120433757177;

double c11 = -462.460321366;
double c12 = 23739.9986309;
double c13 = -194504.35292;
```

```
double c14 = 639383.528867;
double c15 = -523748.057636;
double c16 = -2328271.47551;
double c17 = 7562418.53499;
double c18 = -9668295.89504;
double c19 = 5922081.87086;
double c110 = -1432405.52125;
double c21 = 421.443122208;
double c22 = -14560.354925;
double c23 = 53051.4495633;
double c24 = 382763.793582;
double c25 = -3583589.86875;
double c26 = 12243265.3815;
double c27 = -22307970.0156;
double c28 = 22896656.8499;
double c29 = -12483324.8091;
double c210 = 2813311.71633;
double c31 = -248.783804168;
double c32 = 4807.07241098;
double c33 = 13565.1003309;
double c34 = -466407.780832;
double c35 = 2827083.44764;
double c36 = -8469715.15799;
double c37 = 14459588.8962;
double c38 = -14281087.5331;
double c39 = 7596403.59678;
double c310 = -1684002.64482;
double c41 = 126.965580728;
double c42 = -2090.45270574;
double c43 = 1993.17101166;
double c44 = 100706.510396;
double c45 = -687388.808612;
double c46 = 2132412.46959;
double c47 = -3699199.65914;
double c48 = 3688365.22546;
double c49 = -1975122.39296;
double c410 = 440201.446068;
double c51 = -33.5343446156;
double c52 = 601.878586689;
double c53 = -3064.82070658;
double c54 = 71.7954752052;
double c55 = 51780.666659;
double c56 = -209714.899856;
double c57 = 405011.985355;
double c58 = -428310.461566;
double c59 = 238153.698326;
double c510 = -54497.0973336;
double c61 = 3.97454953787;
double c62 = -77.026846469;
double c63 = 541.19105807;
double c64 = -1696.60270972;
double c65 = 1713.45942707;
double c66 = 4019.01019872;
double c67 = -14844.7928004;
double c68 = 19481.0094551;
double c69 = -12107.0794501;
```

```
double c610 = 2966.92804386;
double c71 = -.170806170177;
double c72 = 3.48182859299;
double c73 = -27.7957587743;
double c74 = 113.762064546;
double c75 = -258.750496922;
double c76 = 311.002585218;
double c77 = -123.917993454;
double c78 = -123.480627492;
double c79 = 154.375042114;
double c710 = -48.5083828701;

static double a1 = 205.8889;
static double a2 = 280.930556;
static double a3 = -317.0138889;
static double a4 = 263.194444;

double b1 = .368105523897;
double b2 = -3.6679548875;
double b3 = 46.6000470809;
double b4 = -262.921061996;
double b5 = 732.99536936;
double b6 = -1076.0613489;
double b7 = 797.948078048;
double b8 = -235.903904222;

double al1 = 153.17055346;
double al2 = -11.7705687461;
double al3 = -1.78126355957;
double al4 = .647385455059;
double al5 = -.0719950751898;
double al6 = .00285423950786;
double a11 = 194.793913463;
double a12 = 74.236124188;
double a13 = 9.84103819552;
double a14 = .436843852745;
double a21 = -74.3508283362;
double a22 = -33.2941879809;
double a23 = -4.78866918581;
double a24 = -.225416733476;
double a31 = 13.0175447367;
double a32 = 6.1586564117;
double a33 = .789740337141;
double a34 = .0321510834958;
double a41 = -.90857587517;
double a42 = -.356752691147;
double a43 = .0238067275502;
double a44 = .00495593933952;
double a51 = -.00071863574153;
double a52 = -.0251026383533;
double a53 = -.0191664613304;
double a54 = -.0017014253867;
double a61 = .00195441702983;
double a62 = .00280533348937;
double a63 = .0013899436563;
double a64 = .000116422611616;
```

```
double tc1 = 647.3, tc2 = 405.5;
double pc1 = 221.2, pc2 = 113.5;
double wm1 = 18.015, wm2 = 17.031;
double trf1 = 273.15, trf2 = .01;
static double r = 8.3144;

double tb, tb1, tb2, tb3, tb4, tb5, tb6, tb7;
double td, td1, td2, td3, td4, td5, td6;
double h_h2ol, h_h2ov, s_h2ol, s_h2ov, v_h2ol, v_h2ov;
double h_h3nl, h_h3nv, s_h3nl, s_h3nv, v_h3nl, v_h3nv;
double he, se, ve;
double Tr, Pr;
```

LIST OF REFERENCES

Alefeld, G., 1989, "Second Law Analysis for an Absorption Chiller," *Newsletter of the IEA Heat Pump Center*, Vol. 7, pp. 54-57.

Barbier, E., 1997, "Nature and Technology of Geothermal Energy: A Review," *Renewable & Sustainable Energy Reviews*, Vol. 1, pp. 1-69.

Benderitter, Y., and Cormy, G., 1990, "Possible Approach to Geothermal Research and Relative Costs," in: Dickson, M. H., and Fanelli, M., eds., *Small Geothermal Resources: A Guide to Development and Utilization*, UNITAR, New York.

Bhatt, M. S., Srinivasan, K., Krishna Murthy, M. V., and Seetharamu, S., 1994, "Absorption-Resorption Heating Cycles with the New Working Pairs R21-NMP and R21-DMA," *Energy Conversion Management*, Vol. 35, pp. 443-451.

Dickson, M. H., and Fanelli, M., 1995, *Geothermal Energy*, John Wiley & Sons, New York.

Drbal, L. F., Boston, P. G., Westra, K. L., and Erickson, R. B., 1996, *Power Plant Engineering*, Chapman & Hall, New York.

Edgar T. F., and Himmelblau D.M., 1988, *Optimization of Chemical Processes*, McGraw-Hill, New York.

El-Sayed Y. M., and Tribus M., 1985, "A Theoretical Comparison of the Rankine and Kalina Cycles," *ASME Special Publication*, AES-Vol. 1, pp. 97-102.

Fiacco, A. V., and McCormick, G. P, 1968, *Nonlinear Programming*, Wiley, New York.

Floudas, C. A., 1995, *Nonlinear and Mixed-Integer Optimization*, Oxford University Press, Oxford.

Goswami, D. Y., 1995, "Solar Thermal Power: Status of Technologies and Opportunities for Research," *Heat and Mass Transfer 95, Proceedings of the 2nd ASME-ISHMT Heat and Mass Transfer Conference*, Tata-McGraw Hill Publishers, New Delhi, India, pp. 57-60.

Goswami, D. Y., 1998, "Solar Thermal Power Technology: Present Status and Ideas for the Future," *Energy Sources*, Vol. 20, pp. 137-145.

Goswami, D. Y., and Xu, F., 1999, "Analysis of a New Thermodynamic Cycle for Combined Power and Cooling Using Low and Mid Temperature Solar Collectors," *Journal of Solar Energy Engineering*, Vol. 121, pp. 91-97.

Goswami, D.Y., Kreith, F., and Kreider, J. F., 2000, *Principles of Solar Engineering*, Taylor & Francis, Philadelphia.

Haenel, R., Rybach, L., and Stegena, L., 1988, "Fundamentals of Geothermics," in: Haenel, R., Rybach, L. ,and Stegena, L., eds., *Handbook of Terrestrial Heat Flow-density Determination*, Kluwer Academic Publishers, Dordrecht.

Hansan, A. A., and Goswami, D. Y., 2001, "Second Law Analysis of Ammonia-Water Power and Refrigeration Thermodynamic Cycle Driven by a Solar Heat Source," *Book of Abstracts*, pp. 115, 2001 Solar World Congress, Adelaide, Australia, International Solar Energy Society, Freiburg, Germany.

Hochstein, M.P., 1990, "Classification and Assessment of Geothermal Resources," in: Dickson, M.H., and Fanelli, M., eds., *Small Geothermal Resources: A Guide to Development and Utilization*, UNITAR, New York.

Hwang, C. L., Williams, J. L., and Fan, L.T., 1972, *Introduction to the Generalized Reduced Gradient Method*, Institute for Systems Design and Optimization, Manhattan.

Ibrahim O. M., and Klein S. A., 1996, "Absorption Power Cycles," *Energy* (Oxford), Vol. 21, pp. 21-27.

Kalina A. I., 1983, "Combined Cycle and Waste-Heat Recovery Power Systems Based on a Novel Thermodynamic Energy Cycle Utilizing Low-Temperature Heat for Power Generation," ASME Paper 83-JPGC-GT-3.

Kalina A. I., 1984, "Combined Cycle System with Novel Bottoming Cycle," *ASME J. Eng. Gas Turbines and Power*, Vol. 106, pp. 737-742.

Kalina, A., Tribus, M., and El-Sayed, Y. M., 1986, "A Theoretical Approach to the Thermophysical Properties of Two-Miscible-Componet Mixtures for the purpose of Power-Cycle Analysis," *American Society of Mechanical Engineers (Paper),* 8p 86-WA/HT-54.

Kalina, A. L., and Leibowitz, H. M., 1987, "Applying Kalina Technology To a Bottoming Cycle for Utility Combined Cycles," *American Society of Mechanical Engineers (Paper)*, 6p 87-GT-35.

Kalina A. I., and Tribus M., 1990, "Advances in Kalina Cycle Technology (1980-1991): Part I Development of a Practical Cycle," & "Advances in Kalina Cycle Technology (1980-1991): Part II Iterative Improvements," *Energy for the Transition Age, Proceedings of the Florence World Energy Research Symposium*, Firenze, Italy, pp. 111-124.

Krakow, K. I., 1991, "Exergy Analysis. Dead-State Definition," *ASHRAE Transactions*, pt 1, pp. 328-336.

Lasdon, L. S., Waren, A. D., Jain, A., and Ratner, M., 1978, "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 4, pp.34-50.

Lee, S. F., and Sherif, S. A., 2000, "Second Law Analysis of Multi-Stage Lithium Bromide/Water Absorption Heat Transformers," *ASHRAE Transactions*, Vol. 106, pp. 105-116.

Leibowitz, H., and Mirolli, M., 1997, "First Kalina Combined-Cycle Plant Tested Successfully," *Power Engineering*, Vol . 101, pp. 44-48.

Lorenz, V. H., 1894, "Die Ausnützung der Brennstoffe in den Kühlmaschinen," *Zeitschrift für die gesammte Kälte-Industrie*, Vol. 1, pp. 10-15.

Maloney, J. D., and Robertson, R. C., 1953, "Thermodynamic Study of Ammonia-Water Heat Power Cycles," ORNL Report CF-53-8-43, Oak Ridge, TN.

Marston C. H., 1990a, "Parametric Analysis of the Kalina Cycle," *J. Eng. for Gas Turbines and Power*, Vol. 112, pp. 107-116.

Marston C. H., 1990b, "A Family of Ammonia-Water Adjustable Proportion Fluid Mixture Cycles," *Proceedings of the 25th Intersociety Energy Conversion Engineering Conference*, Vol. 2, pp. 160-165.

Milora S. T., and Tester J. W., 1976, *Geothermal Energy as a Source of Electric Power Thermodynamic and Economic Design Criteria*, The MIT Press, Cambridge.

Mock, J. E., Tester, J. W., and Wright, P. M., 1997, "Geothermal Energy from the Earth: Its Potential Impact as an Environmentally Sustainable Resources," *Annual Review of Energy and the Environment*, Vol. 22, pp. 305-356.

More, J. J., and Wright, S. J., 1993, *Optimization Software Guide*, Society for Industrial and Applied Mathematics, Philadelphia.

Muffler, P., and Cataldi, R., 1977, "Methods for Regional Assessment of Geothermal Resources," *Geothermics*, Vol. 7, pp. 53-89.

Nocedal, J., and Wright, S. J., 1999, *Numerical Optimization*, Springer, New York.

NREL, 1997, "Geothermal Energy: Power from the Depths," DOE/GO-10097-518.

Park Y. M., and Sonntag R. E., 1990, "A Preliminary Study of the Kalina Power Cycle in Connection with a Combined Cycle System," *Int. J. of Energy Res.*, Vol. 14, pp. 153-162.

Phair, K. A., 1994, "Getting the Most Out of Geothermal Power," *Mechanical Engineering*, Vol 116, pp. 76-80.

Rogdakis, E. D., and Antonopoulos, K. A., 1991, "A High Efficiency NH_3/H_2O Absorption Power Cycle," *Heat Recovery Systems*, Vol. 2, pp. 263-275.

Wark, K., Jr., 1988, *Thermodynamics*, McGraw-Hill, New York.

Winter, C. J., Sizmann, R. L., and Vant-Hull, L.L., 1991, *Solar Power Plants*, Springer-Verlag, New York.

Xu, F., 1996, *Analysis of a Novel Combined Thermal Power and Cooling Cycle Using Ammonia-water mixture as a Working Fluid*, Ph.D. Dissertation, University of Florida.

Xu, F., and Goswami, D. Y., 1999, "Thermodynamic Properties of Ammonia Water Mixture for Use in Power Cycles," *Energy* (Oxford), Vol. 24, pp. 525-536.

BIOGRAPHICAL SKETCH

Shaoguang Lu was born on December 22, 1970, in Linhai, China. He obtained his bachelor's degree in the Department of Thermal Engineering, Tsinghua University, in June 1993. Due to his strong interest in scientific research, which had been cultured since his childhood, he entered the Chinese Academy of Sciences to pursue his graduate study and obtain his master's degree in June 1996. After his graduation, his continued interest in scientific research and a desire to see a different world brought him to the University of Florida. Thus, he started his long journey in August 1996 toward the degree of Doctor of Philosophy.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

D. Yogi Goswami, Chairman
Professor of Mechanical
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

S. A. Sherif
Professor of Mechanical
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Zhuomin Zhang
Associate Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jill E. Peterson
Associate Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Ulrich H. Kurzweg
Professor of Aerospace Engineering,
Mechanics and Engineering Science

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

May, 2002

Pramod P. Khargonekar
Dean, College of Engineering

Winfred M. Phillips
Dean, Graduate School